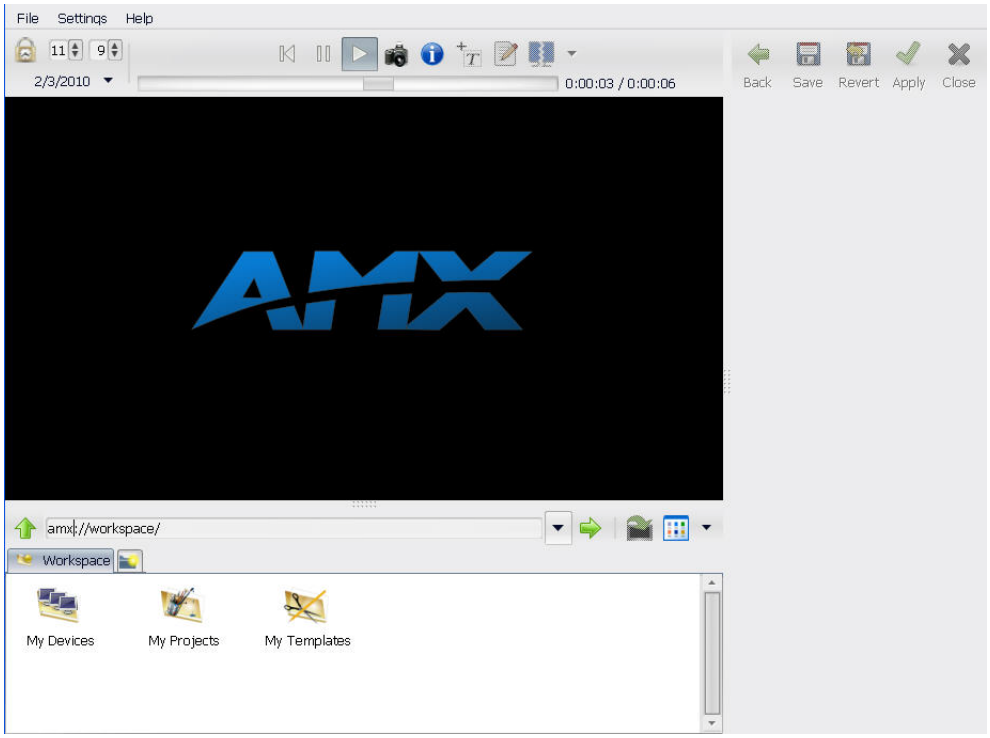
Programming Guide

Inspired Signage XPress

IS-SPX-1000/1300



AMX DOMESTIC CHANNEL PARTNER and END CUSTOMER LIMITED WARRANTY, DISCLAIMER AND LICENSE

(Excerpt from CHANNEL PARTNER TERMS AND CONDITIONS Versions 11.17.2011 with updates for previous version 8.25.2010 [sections 6.1 (a), (b) and (f)])

Definitions

"End Customer" means an authorized end customer with direct in warranty privileges from AMX. Within this limited warranty, disclaimer and license document, "End Customer" shall have the same meaning as "Channel Partner" with the noted exceptions of Sections 6.5 through 6.9 which are not applicable or available to End Customer's directly from AMX. Offerings described in Sections 6.5 through 6.9 are available to End Customer only through their selected authorized AMX Channel Partner.

6. LIMITED WARRANTY; RETURN, REPAIR AND REPLACEMENT

6.1 AMX warrants the Products to be free of material defects in materials and workmanship under normal use for three (3) years from the Shipping Date (or such other period as may be specified below), subject to the following limitations and exceptions ("Limited Warranty"). For any Product, "Warranty Period" means the period during which the Limited Warranty is in effect, as set forth herein.

(a) LCD and LED panels are warranted for three (3) years from the Shipping Date, except for the display and touch overlay components, which are warranted for a period of one (1) year from the Shipping Date.

(b) Disk drive mechanisms, pan/tilt heads and external power supplies are warranted for a period of one (1) year from the Shipping Date.

(c) AMX lighting Products are warranted to switch on and off any load that is properly connected to our lighting Products, as long as the AMX lighting Products are under warranty. AMX also warrants the control of dimmable loads that are properly connected to our lighting Products. The dimming performance or quality thereof is not warranted, due to the random combinations of dimmers, lamps and ballasts or transformers.

(d) AMX software and firmware included in the Products is warranted for a period of ninety (90) days from the Shipping Date.

(e) Batteries and incandescent lamps are not covered under the Limited Warranty.

(f) The Warranty Period for AMX AutoPatch EPICA, Enova DGX, Modula, Modula Series 4, Modula Cat Pro Series and 8Y-3000 Product models will continue for the original installation until five (5) years after the issuance of a PDN with respect to termination of the applicable Product model. However, if the Product is moved from its original installation to a different installation, the Warranty Period will automatically become three (3) years from the Shipping Date and, if more than three (3) years have elapsed since the Shipping Date, the Warranty Period will automatically expire.

DLI-6293353v1

Note: *Refer to www.amx.com to view/download the latest complete AMX Warranty and Return Policies.*

Table of Contents

Inspired Signage XPress Programming 1
Introduction 1
What Can You Do With XPress? 1
When to use XPress Authoring Features? 1
Installation 1
Installing Inspired Signage XPress for the First Time 1
Minimum Recommended PC Requirements 2
Installation Location 3
XPress Removal 3
Inspired Signage XPress UI 3
Learning XPress Authoring Features 4
XPress Projects 4
Creating a New Project 5
Browsing Panel 6
Importing Multi-Media Files 7
Importing Images 7
Importing Video 7
Importing PowerPoint Presentations 8
Importing Flash (SWF) Files 8
Creating and Editing SVG Files 8
Editing an Existing SVG Document 9
Edit a Media Property 10
Interactive Editing 11
Interactively Editing SVG Files 11
Editing Media Elements Layout 12
Editing Media Links 13
Editing Text 13
Multi-Layered SVG Templates 14
Composing a Scene 14
Customizing a Multi-Layer SVG Document 14
Example: Editing 15
Playlist Template 15
Composing a playlist SVG document 16
How to Customize a Playlist SVG Document 16
Special News Templates 18
Composing a Live News Document 18
Customizing a News SVG Document 19
news.svg 19

news_ticker.svg 19
news_crawler.svg 19
Using Format Helpers 20
bouncing_text.svg 20
fit_text.svg 20
date_time.svg 20
switch.svg 20
Scalable Content and Aspect Ratio of Your Display 21
Multimedia Effects 23
Adding Effects 23
Configuring Effects 23
Previewing 24
Preview XPress Content 24
Publishing 25
Using Quick Publish 25
Monitoring 26
Manually Add Devices to XPress 26
Check the Status of the Devices 27
Open a Device 27
Scheduling 27
Scheduling the Content Displayed by an IS-SPX Player 27
Create a New Schedule 28
Open a Schedule 28
Add Events To a Schedule 29
Modify the Scheduling Properties of an Event 30
Define Default Events and Recurrences 30
Define a Default Animation 30
Define Recurring Events 31
Preview Scheduled Content 32
Data Polling Rates 32
Remote Control 33
Overview 33
Remote Control of Content 33
When To Use Remote Control 33
How To Use Remote Control 33
Triggering a Content Switch Over IP 35
Example: Testing 36
Switching Content With the Serial Port 37
Switching Content With a USB HID Device 37
Switching Content From Another Device 38

Remote Control of Playback 39
When To Use Remote Control of Playback 39
How To Use Remote Control of Playback 39
Interactive Pause/Resume Over TCP 40
Interactive Pause/Resume Using HTTP GET 40
GET Embedded In an HTML Page 41
Interactive Pause/Resume From a Device 41
Timed pause/seek/resume 41
Getting Feedback On the Playback Progress 42
Knowing the Current Position With Extra Accuracy 42
Remote Triggers 43
Triggering a Text or Image Overlay 43
Triggering an Alternative Content 43
Triggering of SVG Animations 44
Synchronized Content 45
Overview 45
Application Examples 45
Multiscreen Displays 45
Designing Basic Projects for Multiscreen Layouts 45
Creating a Configurable Multiscreen Project 46
Creating a Multiscreen Project from Templates 48
Editing Multiscreen Content 50
Configuring XPress and the IS-SPX Player 51
Create a Virtual Device 51
Configure the Multiscreen ID 51
Publish Your Project To Your Device 52
Synchronization 52
Synchronizing Devices 52
Configure NTP Through the HTTP Interface 53
Advanced Configuration 54
Using an IS-SPX Player as an NTP Server 54
Configuration 54
Configuration of the IS-SPX Players Acting As Slaves 54
Configuration of the IS-SPX Player Acting as Master 55
Master/Server Architecture Examples 55
Syntax for Multiscreen Synchronized Content 56
Canvas and Viewbox Concepts 56
Multiscreen Elements 56
Synchronization Elements 56
The iCalendar (.ics) File 56
The spx:begin Element 57

Additional Syntax for Multiscreen Content 57
Extension of the SVG <switch> Element 57
Extension of JavaScript 58
Example: Screen Layout With Six Screens 58
Example: Screen Layout With an Empty Border Between the Screens 58
Example: Screen Layout With "Spread Displays" 59
Example: <switch> SVG Element 59
Example: JavaScript 60
Troubleshooting 60
NTP Synchronization 60
Project Creation 60
Internal Clock Calibration 60
Interactivity 63
Overview 63
Creating Interactive Content 63
Interactivity Template Files 63
Creating a Touch Screen Menu 64
Creating Buttons 64
Controlling Media 65
Testing the Project 66
Creating a Keyboard Menu 67
Creating the Menu 67
Configuring the menu 68
Controlling the Menu 70
Keystrokes 71
Controlling Media 72
Testing the Project 73
Suggestions for Further Projects 73
Playback Projects 73
Playback Template 74
Controlling the Media 75
Playback Position 75
Example: Media player 75
IS-SPX Player Configuration 78
Calibration For Touch Screens 79
Calibration of a Touchscreen Through the IS-SPX Player 79
Manual Calibration Procedure 80
Troubleshooting 80

Serial Protocol File Editor 81
Overview 81
Adding a List Variable 81
Adding a Range Variable 82
Exporting the Finished File 83
Shared Variables 85
Overview 85
Concepts Behind Shared Variables 85
Javascript API 85
Interface 85
Sample Code 86
Setting the Variable 86
Reading the Variable 86
Network API 86
Framing 86
List of Commands 87
UPDATE 87
EVENT 87
SUBSCRIBE 87
UNSUBSCRIBE 87
Enabling the Network API 88
API Emulation 89
Appendix 91
Serial Control 91
Serial Control From Within an Existing Inspired Signage Project 91
Gathering Display Serial Protocol Commands 91
Creating a Serial Protocol File 92
Creating the Power variable: 92
Uploading the Serial Protocol File 96
Creating New Templates 97
Creating and updating a new calendar file 98
Serial Control From NetLinx 100
Adding Additional Templates to Inspired Signage XPress 100
Commonly Used Terms 101
FSM 101
SVG 101
URI 101

Inspired Signage XPress Programming

Introduction

The Inspired Signage XPress media content developer (referred to hereafter as "*XPress*") allows users of the IS-SPX-1000 and IS-SPX-1300 IS-SPX Players to manage media elements in a digital signage application. Its purpose is to combine and customize media elements created in different applications in order to build digital signage content of professional quality.

What Can You Do With XPress?

- Combine an image background with text, videos, animations, and graphics.
- Interactively customize imported media elements, such as controlling positions in the scene, font properties, color, opacity and transparency.
- Interactively add and edit text with sophisticated layout.
- Change the relative order of each media element in the scene to create depth effects.
- Apply animation effects on most media appearance properties.
- Create ordered playlists of media elements of different nature.

Due to the size of the internal storage space in the IS-SPX Player, the IS-SPX Player cannot display files larger than 2 gigabytes in size if these are loaded into the device's memory. However, larger files and projects may be displayed if these are stored in an attached USB storage device, such as a portable hard drive or USB thumb drive.

When to use XPress Authoring Features?

- You are managing the digital signage of a restaurant. You already have the pictures of your special menus, your logo and a few advertising video clips. XPress lets you import and position all of these media on the same slide. You can then type your menu list, customizing fonts and colors. In a few clicks, you have created or updated your menu of the day.
- You are managing the digital signage of a pharmacy. You already have the pictures of your products and the corresponding advertising clips. XPress lets you import and position these media on the same slide. It lets you also create a playlist of all the products you want to highlight.
- You are managing the digital signage of a hotel resort. You already have advertising videos and images of your establishment and your services. You have created power point presentations welcoming special guests or presenting the upcoming events organized in your resort, and you have high quality images of the major city spots. XPress lets you import all of these individual media and order them into a single playlist. It is very easy to further reinforce your graphical identity by adding your logo, your slogan, a background, or a graphical layout on each slide of the playlist. In a few clicks and without special graphics skills, you can author your personalized and updated content and you can schedule its distribution on the right display and at the right time.
- You are a graphical professional, and you are using a particular software solution to author new media files. These can be exported as images, videos, vector graphics content, or animations. You will be able to import them later in XPress, combine them with additional elements and customize the final canvas for a specific application.

Installation

Installing Inspired Signage XPress for the First Time

The 30-day trial of Inspired Signage XPress is available from **www.amx.com**. A license for the full version of either XPress Standard or XPress Pro may be purchased from AMX at any time before or after the expiration of the trial version.

- **XPress Standard** allows control of up to 10 IS-SPX Players at one time.
- **XPress Pro** is intended to allow control of up to 100 IS-SPX Players at one time.

If you need control of more than 100 IS-SPX Players at one time, please contact AMX for more information.

Minimum Recommended PC Requirements

IS-XPress will operate on Windows XP, Vista and Windows 7 (both 32-bit and 64-bit) PCs that support of DirectX 9.0c and SSE2 instructions. For best performance, AMX recommends a minimum of a Pentium D processor with separate graphics card. IS-XPress may operate on older PC platforms, but less responsively. Also, please note that other software applications can also degrade the responsiveness of IS-XPress.

NOTE

Due to potential issues with license validation, Inspired Signage XPress should not be installed on a virtual machine.

NOTE

Both the Standard and the Pro Licenses only apply to one computer installation of the software at a time. If the software is installed on a second computer, a new license must be purchased for that computer.

After installing the software, open XPress. This opens the *Software License Validation* window (FIG. 1).

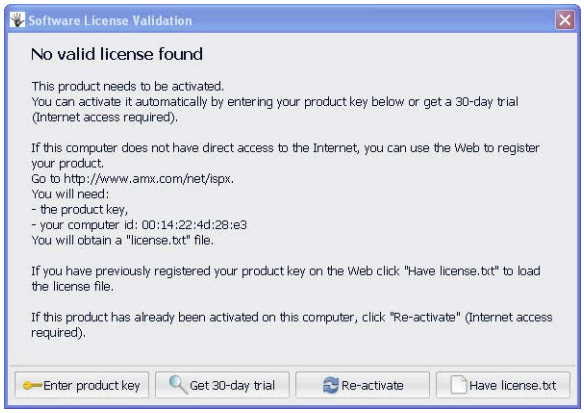

Software License Validation
No valid license found
This product needs to be activated.
You can activate it automatically by entering your product key below or get a 30-day trial (Internet access required).
If this computer does not have direct access to the Internet, you can use the Web to register your product.
Go to http://www.amx.com/net/ispx.
You will need:
- the product key,
- your computer id: 00:14:22:4d:28:e3
You will obtain a "license.txt" file.
If you have previously registered your product key on the Web click "Have license.txt" to load the license file.
If this product has already been activated on this computer, click "Re-activate" (Internet access required).
Enter product key
Get 30-day trial
Re-activate
Have license.txt

FIG. 1 Software License Validation window

A purchase of either an XPress Standard or a Pro license comes with the product activation key. Click the **Enter Product Key** button to open the *Online Product Activation* window (FIG. 2). In the provided fields, enter the key to complete the activation.

FIG. 2 Online Product Activation window

NOTE

XPress must be restarted to complete the activation.

- If you choose not to purchase a license at this time, click the **Get 30-Day trial** button, and restart the application.
- If you have installed a previous version of XPress with a product activation key, click the **Re-activate** button to use the previously used key to activate this version. If the correct key is entered, the application will note that the re-activation was successful.

Installation Location

When installed, XPress is accessible from the Windows Start menu in the *Programs/AMX Control Disc* directory, and this directory will be created if it does not already exist. Open the Inspired Signage directory (FIG. 3) to run XPress.

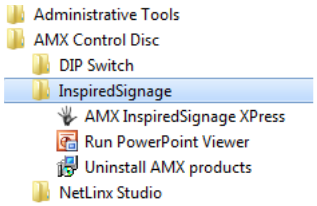


FIG. 3 AMX Control Disk directory with XPress

XPress Removal

If XPress needs to be removed from a computer for any reason, it must be removed via the *Uninstall AMX products* utility. In particular, any upgrade to a new version of XPress requires that the previous version be uninstalled. In the *Programs/AMX Control Disc/Inspired Signage* directory (FIG. 3), select *Uninstall AMX products* to remove the application from the computer.

WARNING

If you intend to update to the latest version (v2.2.6) of XPress, you need to upgrade all IS-SPX Players in your network simultaneously with firmware v2.2.6. Content created with XPress 2.2.6 may not be compatible with previous firmware versions. If you upgrade devices containing an already existing installation, AMX recommends testing your content first on a updated test device.

Inspired Signage XPress UI

The Inspired Signage XPress main interface (FIG. 4) presents a default animation of the AMX logo and the files available in the Workspace area. The controls include time controls, test animation controls, and options to save and revert changes.

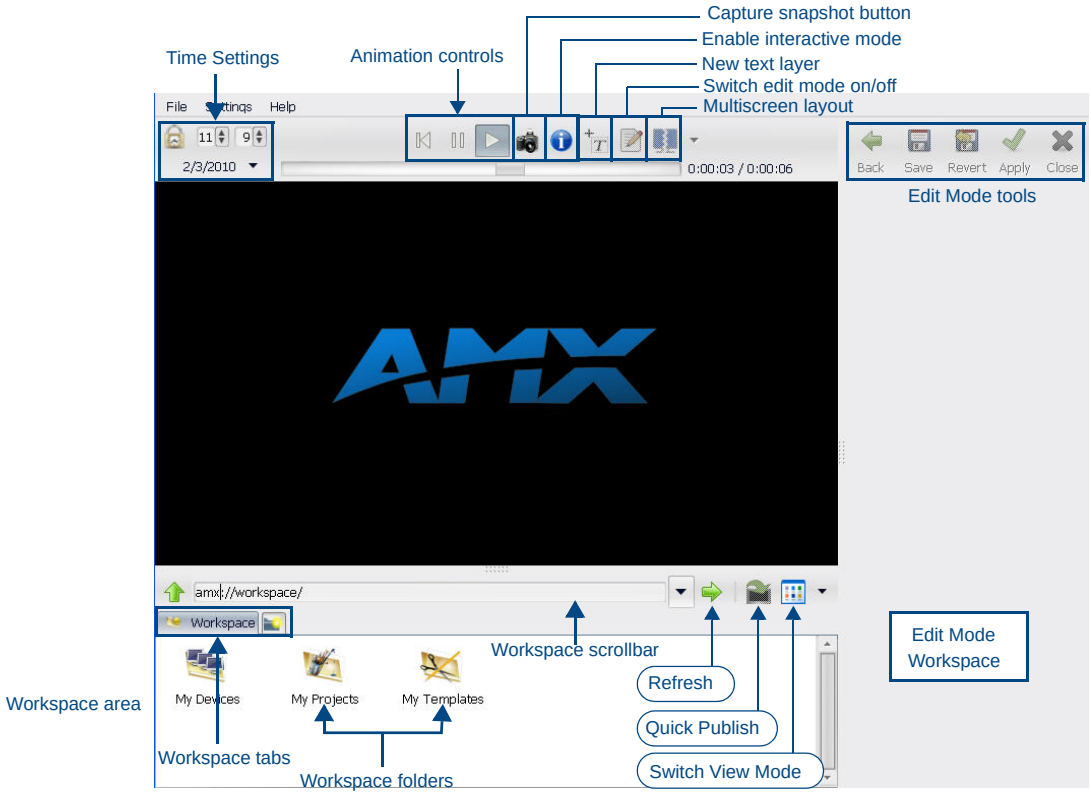


FIG. 4 XPress - Main interface

XPress Main Interface	
Time Settings:	The Time Settings controls set the date and time at which a project will run. Click the **Lock** icon to lock the time to the internal clock in the IS-SPX Player.
Animation Controls:	The Animation Controls determine the direction in which an animation test will run. Click the **Play** button to run a project and the **Pause** button to stop it: the **Restart** button will return the project to its beginning in either *Play* or *Pause* mode.
Capture Snapshot button:	Click the **Capture Snapshot** button to save an image of the currently displayed animation.
Enable Interactive Mode button:	Turns the Interactive Mode on the main viewing pane on and off. This allows you to test any interactive content on which you may be working.
New Text Layer button:	Adds a new text layer on the currently viewed project.
Switch Edit Mode icon:	Turns the Edit Mode on the main viewing pane on and off.
Multiscreen Layout button:	Displays multiple panes for projects that utilize layout displayed on multiple screens.
Edit Mode Tools buttons:	These buttons (**Back**, **Save**, **Revert**, **Apply**, and **Close**) are only functional when the main viewing pane features an item being displayed in Edit Mode.
Workspace Scrollbar:	The Workspace Scrollbar displays the currently running project. Click the arrow to the left of the scrollbar to return to the previously opened project and click the arrow to the right of the scrollbar to show the last opened projects.
Refresh button:	This button refreshes all of the current content in the chosen project tab.
Quick Publish button:	This button allows the currently selected project to be published directly to a selected IS-SPX Player.
Switch View Mode icon:	The dropdown menu to the right of the icon allows the content in the currently selected project to be displayed as icons (**Icons**) or with details on the file content such as size, type, and date last modified displayed (**Details**).
Workspace Tabs:	These tabs display the projects or other content already open. To open a new project, click the folder icon to the right of the last open project.
Workspace Area:	This is the main area used to display content within a project or Template. Drag and drop icons or details from the Workspace area onto the main viewing pane.

Learning XPress Authoring Features

The first step towards using XPress authoring features is to understand the concept of an XPress project. Read the section devoted to this subject in the *XPress Projects* on page 4.

On these projects, you may experiment the main authoring functionalities such as:

- Operations on files and collections
- Creating new documents
- Editing a document
- Editing a multi-layer document
- Editing a playlist
- Managing the aspect ratio

XPress Projects

A *Project* is any collection of author and preview content intended to be published later on IS-SPX Players. A project is a special folder that contains a master file (index.svg) and all the media elements (e.g. video clips, images, texts, animations, scripts, etc.) that are composed together to create the XPress presentation. Note that not all media elements need to be copied inside a project, and IS-SPX Player can be instructed to access remote locations and retrieve content on the fly by using PHP scripts. This structure is very similar to a web page with an index.html file and all the media elements that will appear when the user browses the web site.

By default, XPress regroups all the projects under *My Projects*, a folder automatically created on your hard drive while installing XPress under **\\My Documents\AMX\Xpress\My Projects**. The folder *My Projects* appears in the browsing area when you select the default Workspace view.

When you open the *My Projects* folder (FIG. 5), you can access a number of actions by right-clicking the browsing area. In particular you can:

- Create new projects.
- Create new collections of projects so as to organize your projects in functional groups.

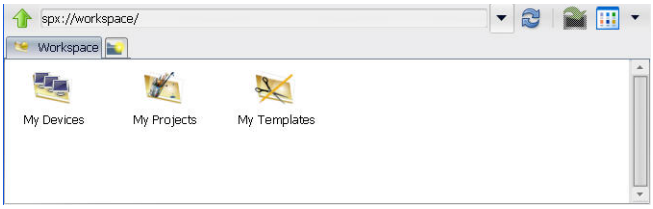


FIG. 5 My Projects folder

The notion of a project as introduced by XPress has several advantages. In particular:

- The user can create projects offline, preview them and keep modifying them without affecting the actual presentations displayed by an IS-SPX Player. When the project is finalized, it is easy to publish it on the target device.
- The user can apply conventional backup systems on the *My Projects* folder to reduce the risk of losing data.
- A single project can be published on a single player or multiple players, which reduces the effort required to manage the same content on different displays.
- When the network connection between the PC and the IS-SPX Players is slow or temporarily unavailable, XPress offers an offline solution to edit the project content.
- You can open multiple views and navigate inside each of them: each view will be displayed as a different tab.

Creating a New Project

To create a new project:

1. Right-click in the *Workspace* under the *My Projects* tab. This opens the *New Project* window (FIG. 6).

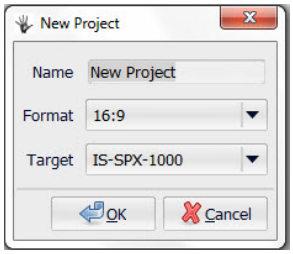


FIG. 6 New Project dialog - Name

2. Choose a name for the project and enter it in the *Name* field.
3. Enter a particular format from the *Format* dropdown menu (FIG. 7).

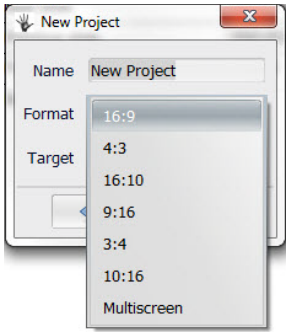


FIG. 7 New Project dialog - Format

NOTE

The Multiscreen format uses the configurable.svg file from the Multiscreen template folder. For more information on multiscreen formats, please refer to the Synchronized Content section on page 45.

4. Select either IS-SPX-1000 (default) or IS-SPX-1300 as the target device for the project, from the *Target* dropdown menu. This selection ensures that created or imported content for this project is within specification for the selected Player (FIG. 8):

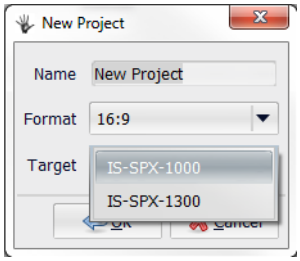


FIG. 8 New Project dialog - Target

5. Click **OK** to create the new project folder.

Browsing Panel

The browsing panel provides a few basic file operations through a pop-up menu that appears when right-clicking your mouse. The available operations change depending upon the area clicked:

If you right-click directly in the browsing panel of a specific view (make sure that no file or folder is currently selected), the following functions appear whose actions apply directly to the specific view currently opened:

- New SVG file: Create a new SVG multilayer file. Select the aspect ratio of the new document from 16:9, 4:3, 16:10, 9:16, 3:4, or 10:16.
- New Playlist: Create a new Playlist file.
- New Schedule: Create a new scheduling file (with extension .ics).
- New Collection: Create a new folder.
- Upload...: Upload a document from your hard drive. Alternatively, files may be uploaded by dragging-and-dropping it from a Windows Explorer window.

If you right-click a specific file (or you right-click the browsing panel while a file has been selected), the following functions appear whose actions apply directly to the selected file:

- Open: Opens the file in the preview panel.
- Delete: Deletes the file without confirmation (short cut: shift+delete)
- Rename: Renames the file.
- Update metadata: Updates the icon of the file according to its content.
- View source: View source code if allowed on the specific file. If not this will be disabled
- New SVG: Create a New SVG file
- New Playlist: Create New Playlist
- New Schedule: Create new Schedule
- Upload: Upload a file to the project.
- Cut / Copy / Paste
- Up: Up a folder level
- Refresh: Refresh the current view.

If you right-click a Collection (Collections are equivalent to Folders in Microsoft Windows and can be used to arrange your media in a hierarchical manner) in addition to the above functions, you can access the *Set type* menu. This menu allows you to specify what type of collection you want (FIG. 9).

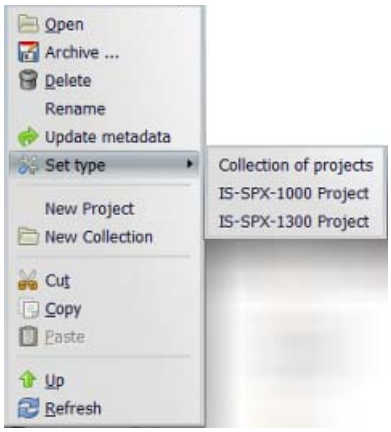


FIG. 9 Set Type menu

Changing the type of a Collection modifies the behavior of XPress when opening the Collection.

The browsing panel allows the user to copy files and folders with intuitive drag and drop operations. For example copying a file from its position to a new destination can be done as follows:

1. Open a browsing view where your original file is located. To locate the original file, you can navigate the directory path from an existing view or directly type the URL in the address area of the browsing panel.
2. Open a new browsing view where your copy destination is located.
3. Drag and drop the file from the original position to the new view. Pressing *shift* during the copy will move the element instead of copying it.

The same mechanism applies for collections and groups of files and collections.

Importing Multi-Media Files

XPress lets the user import many types of media files into an XPress project (or other folders) from a standard Microsoft Windows Explorer window. The operation is as simple as a drag-and-drop from the Explorer window to the target XPress browsing area. You can also use the special *upload...* command, accessible by right-clicking the browsing area of XPress.

The advantage of importing media using the above procedures is that XPress can modify or transform the media automatically so as to guarantee its content will be easily displayed easily displayed within the boundaries of the IS-SPX-1000 or IS-SPX-1300 Player. Whenever possible, use the above procedures when building your XPress project.

Importing Images

Importing images into XPress will ensure that they can be properly displayed by the IS-SPX Player.

- Images too large will be resized.
- Images in formats not supported by the IS-SPX Player (such as *.bmp*) will automatically be converted to PNG images. If an image called *example.bmp* is uploaded, it will be renamed to *example.bmp.png*.

Importing Video

Video can be imported into XPress. During the import process, XPress will check that the imported video can be played by the IS-SPX-1000 or IS-SPX-1300 Player. Otherwise, XPress will not import the video, and a message will be displayed to inform the user. In this case, check that the CODEC used to encode your video are supported by the IS-SPX Player, and that the video resolution is within the IS-SPX Player specification.

Please see the IS-SPX-1000 & IS-SPX-1300 Operation Reference Guide for information on encoding videos.

Once the video has been imported, you can preview it directly using XPress, by double-clicking on the video. By default, the preview of the video corresponds to the first frame of the imported video. You can change the preview of the video in the browsing area by using the **Capture Snapshot** button (see FIG. 4 on page 3).

Importing PowerPoint Presentations

While the IS-SPX Player doesn't support direct display of PowerPoint presentations (PPT). it does provide import functionality for a PPT, so that when a PPT is imported into an XPress project, it is immediately converted to video (without audio).

NOTE

Microsoft Office 2010 is not used by the import mechanism. If you have pre-installed Microsoft Office 2010 then you must manually install PowerPoint viewer 2007.

When a PPT file is imported to XPress, the conversion status is displayed in an new dialog. Depending on your presentation, this conversion can take up to the duration of the presentation. At the end of the conversion the resulting video will contain all PPT slides with all animations.

NOTE

PPT files are converted using a ppt2svg.exe program.This conversion removes the audio.

For the conversion to work properly, the PPT file should contain automatic timings to control slide transitions.

The speed and the quality of the importing process can be modified using the Settings menu. The import quality lets you choose between the following:

- **720p:** HD-ready quality at 15 fps. Use this settings when converting a presentation with small amount of animations.
- **High:** SD resolution at 30 fps. This mode provides improved quality for small size text. If your presentation includes fonts smaller than 24 pt and many animations, it is recommended to use the high quality mode.
- **Fastest:** SD resolution (480p at 30fps for either MPEG-2 or MPEG-4)

Importing Flash (SWF) Files

The IS-SPX Player doesn't support the direct display of flash files (SWF). it does provides import functionality for SWF files. When an SWF is imported into an XPress project, it is directly converted to video.

NOTE

You need to have the Flash 9 or 10 ActiveX control installed on your PC for this conversion to work. If this is not the case, please install it from Adobe's web site.

Creating and Editing SVG Files

The simplest way to create an SVG document is to right-click the browsing panel and select the action *New SVG file* with the desired aspect ratio. XPress creates a new SVG multilayer document *new.svg* in the current browsing view with the selected aspect ratio (FIG. 10).

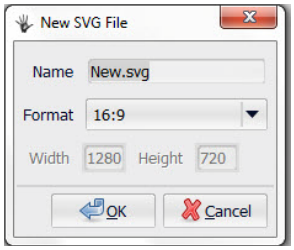


FIG. 10 Creating a new SVG file

Having installed XPress, you can also benefit from a set of pre-defined SVG documents called *templates*. You can find these templates under the collection *My Templates* in the default Workspace view. Provided templates include:

- Multilayer templates with different aspect ratios
- Text templates
- Playlists templates
- Animated Clocks (analog and digital) templates
- Templates for displaying RSS data from a web site

Other templates will be provided in future releases of XPress. For more information, please refer to the *Adding Additional Templates to Inspired Signage XPress* on page 100.

To create a new SVG document based on one of these templates, simply copy (or drag and drop) the one you have selected from the *My Templates* collection to your target project.

Editing an Existing SVG Document

The first step required to edit an SVG document is to open it: either right-click the document and select *Open*, or simply double-click the file.

If possible, XPress will start previewing the content of the document in the Preview panel, and the Editing panel will change according to the content of the document you have open. In fact, XPress can recognize specific tags and activate appropriate editing interfaces to help the user customize the document without any knowledge of SVG syntax.

- If the document you have opened was not designed to activate special editing interfaces, then the editing panel will show only the *XML Tree* and *WebDAV* tabs. You can still edit the document content using the *XML Tree* tab interface, but you need to have an understanding of XML structure and SVG syntax.
- If the document you have opened was designed to activate special editing interfaces then an additional *Properties* tab will appear. This tab gives access to a number of parameters that can be customized directly from the *Properties* tab. Changing these parameters will directly modify the XML Tree structure of the document, but you do not need to know SVG syntax to do it.

Each template document can expose one or multiple parameters. Parameters can be of the following types:

- **Value:** The user can type some values to modify the layout of the document. Generally those are durations, sizes or number of items to display.
- **Text:** Modify some text shown on the screen. The user can also select the fonts and size of the text.
- **Color:** Choose the color of an element, like the background for instance.
- **Choice:** Choose between a set of values for the property.
- **Edit Media:** Select a media. The media can be modified by dragging any media from the browsing panel. Additionally, properties such as duration, number of repetitions of the media, and URL can be edited by pressing the "..." button at the bottom right of the property.
- **Edit Text:** Edit a text box. Additionally the user can add a background or a border.
- **Groups:** Any of the above properties can be grouped. Groups might be opened or closed by default. Pressing on the arrow, will close or open the group depending on its current state.

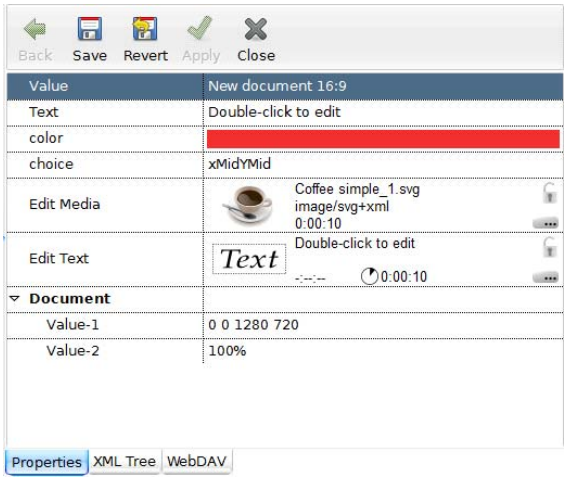


FIG. 11 Template Document Parameters

Edit a Media Property

The *Media* property allows you to control the display properties of the media linked to the current document. This property is used by many templates, i.e. multi-layers and playlists. The media property shows the following information to the user:

- *The media URL.* (e.g. video.avi or the absolute URL) If the media is located in the same project as the document, the URL is relative.
- *The media type.* (e.g. image/spectral) Indicate the type of media as detected by XPress.
- *The total playing time of the media.* A looping symbol is added to the bottom right of the column if the media loops at least once.
- *The starting time.* A timer clock with the delay value is displayed if the media starting time is different than zero.
- *A star.* This indicates that multimedia effects are applied to this media.

The properties of the media can be edited using the "..." at the bottom right of the media property. This will open the media description dialog (FIG. 12).

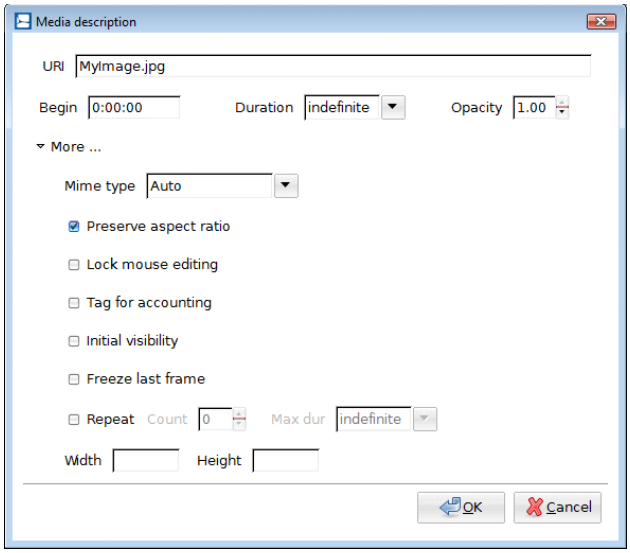


FIG. 12 Media Description window

The media description dialog lets you change:

- URI: The URL of the media. You can type a relative URL (for addressing those media inside your project) or an absolute URL for those media located in remote servers. (For more information on the difference between a URI and a URL, please refer to *Commonly Used Terms* on page 101.)
- Begin: The begin time of the media relative to the beginning (the opening) of the document. By default all media begin as soon as the master document is opened.
- Duration: The duration of the media. You can choose either to enter an absolute duration (enter a time in the following form **hours:minutes:seconds**) or to select Media to force the player to use the duration of the media itself, or to select indefinite to force the media to play for an indefinite period of time. If XPress is able to compute the duration of the media, it will pre-set the absolute duration.

  ***Note**: If you specify an indefinite duration for your media, it will NOT repeat. If you want your media to repeat forever, you should check Repeat with a Count of 0 and an indefinite maximum duration. If both a Count and a Max duration are specified, the media will repeat until the minimum of the two is reached.*
- Opacity: The opacity of the media. A value equal to 1, means that the media is opaque. A value equal to 0, means that the media is completely transparent.

By default the following parameters are not shown. To make them visible, expand the *More...* item.

- Lock mouse editing: When selected the interactive editing for this media is disabled.
- Mime type: The media type. This should be detected correctly by XPress and should not need to be modified.
- Preserve aspect ratio check box: Controls if the aspect ratio of the media should be preserved when included into the master document.
- Tag for accounting: When checked, the opening and closing time of this media will be logged in the accounting log files.
- Initial visibility: When checked, the media will be visible (but not animated) before the beginning of the animation.
- Freeze last frame: When checked, the media will be visible (but frozen) after the end of the animation.
- Repeat: If you want your media to be displayed more than once, select this check box. You can then specify:
  Count: The number of time the media should be repeated or
  Max dur: The duration of the repetition in seconds.
- Width and Height: Lets you control the size of the media.

Interactive Editing

The preview panel has two main functionalities:

- **Preview IS-SPX content and multi media content.** The preview panel emulates in software the display capabilities of the IS-SPX Player. Moreover it offers the user the possibility to interactively pause and seek the content using the timeline at the top of the preview panel itself.
- **Interactively edit XPress content.** XPress content is described with the SVG language (see W3C specification). SVG language follows the XML document structure and thus can be easily edited by hand or with other software applications. To simplify the editing task: XPress offers an intuitive graphical user interface directly through the Preview panel. The user can interact with media elements in the scene by changing their style properties, positions, sizes, rotations, etc... and XPress automatically changes the corresponding SVG document to reflect the changes. This is done in real time, while the XPress content is previewed resulting in an intuitive and powerful SVG editing console.

Interactively Editing SVG Files

Some SVG files have editable properties. This is the case with all the templates (found under *My Templates*), and new documents created by XPress.

Having editable properties means that the SVG document can be interactively customized using XPress. When opened with XPress, these special SVG files will instruct the GUI to open a property Tab in the editing panel. The property tab includes all the fields that have been defined as interactively editable by the author of the SVG document. These fields may be text, media elements or parameters.

Editing Media Elements Layout

Media elements can be declared as transformable in SVG templates. If a media is transformable, then its layout can be edited directly in the preview area. Such elements appear in the preview panel highlighted by a black and yellow border. Simply move your mouse cursor on a media element to check if it can be transformed.

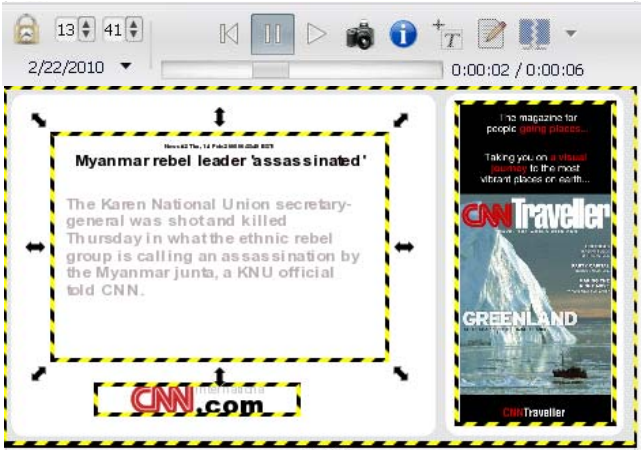

13
41
2/22/2010
0:00:02 / 0:00:06
Myanmar rebel leader 'assassinated'
The Karen National Union secretary-general was shot and killed Thursday in what the ethnic rebel group is calling an assassination by the Myanmar junta, a KNU official told CNN.
CNN.com
CNN Traveller
GREENLAND

FIG. 13 Demonstration of Media Elements

When you select a media element by clicking it, four arrows appear on the corners of the selected media. You may now do the following interactive operations on this media directly with your mouse:

- **Move:** Click on the element to move it using the mouse.
- **Zoom:** Click on one of the arrows to modify the dimension of the element using the mouse.
- **Resize:** Click on one of the arrows and press CTRL to change the size of the element.
- **Rotate:** Click once on the element, the arrows will be modified. Click on the arrow on the corner to rotate the element
- **Skew:** Click once on the element, the arrows will be modified. Click on the top, or side arrows to skew the element

The difference between the zoom and the resize action depends on the elements.

- **Text area:** *Zoom* will change the size of the text. *Resize* will modify the text area, thus enabling to write more text.
- **Media** (image, video, animation)**:** *Zoom* will change the size of the media. *Resize* will change the size of the media area, thus the aspect ratio of the media will not be modified.
- **Rectangles:** *Zoom* will change the size of rectangle including the border. *Resize* will change the size of the rectangle, but not the border.

The image below shows an example of the effect of the Zoom and Resize operation on the different elements.

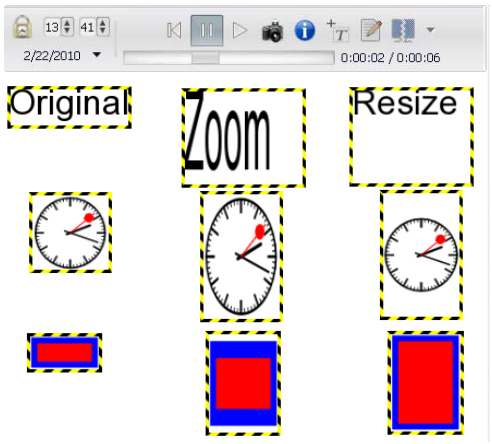


FIG. 14 Effects of Zoom and Resize on Media Elements

The right-click on an element opens a popup menu with the following shortcuts:

- **Full Screen:** Zooms the element so that it occupies the entire document.
- **Rotate Right:** Rotates the element by 90 degree clockwise.
- **Rotate Left:** Rotates the element by 90 degree counter-clockwise.
- **Flip Vertical:** Flips the element upside-down.
- **Flip Horizontal:** Flips the element left-right.
- **Reset Transform:** Removes all zoom, rotation and skew on the element.

The other options are related to multilayer documents and are described in more detail in the *Multi-Layered SVG Templates* on page 14 and the *Multimedia Effects* on page 23.

Editing Media Links

Media links can be declared as editable in SVG templates. When a media from the browsing panel is dragged to the preview panel, editable media link are highlighted in green and a magnet icon is shown on top of the media. When the media is moved on top of an editable link, the highlight is changed to red, and the magnet logo is modified. Dropping a file from the browsing area of such an element will modify the content currently displayed by the player (FIG. 15).

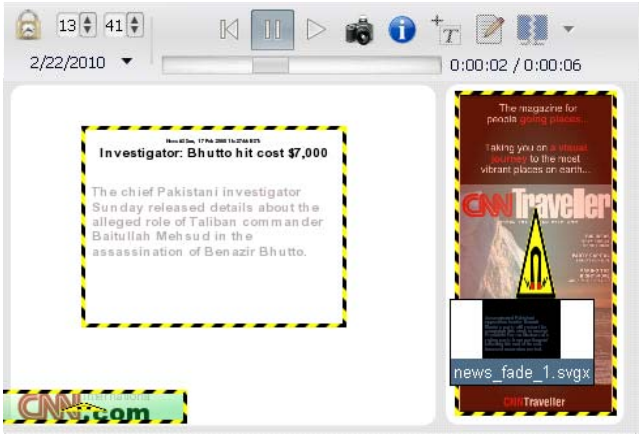

13
41
2/22/2010
0:00:02 / 0:00:06
Investigator: Bhutto hit cost $7,000
The chief Pakistani investigator Sunday released details about the alleged role of Taliban commander Baitullah Mehsud in the assassination of Benazir Bhutto.
news_fade_1.svgx

FIG. 15 Editing Media links

On some documents, such as the multilayer template, dropping a file, with the CTRL key pressed, will not replace the media. This will instead create a new media in the current documents. This action is symbolized by a plus on the mouse cursor.

Editing Text

Text might also be declared as editable. In this case, it is possible to edit its content directly by double clicking its element in the preview panel: a cursor will appear. The menu on the top of the preview panel will let you modify the font style, color and size (FIG. 16).

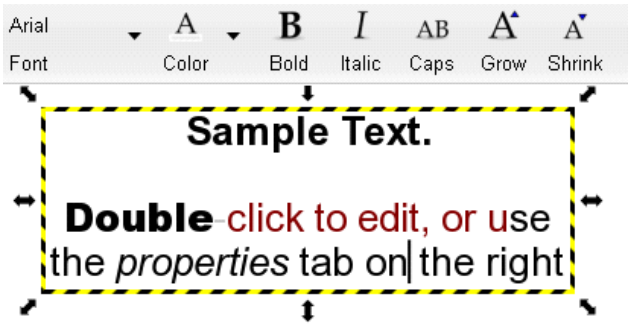


FIG. 16 Editing Text Elements

Multi-Layered SVG Templates

By default, each time you create a new SVG document by right-clicking the browsing panel, selecting *New SVG*, and choosing your aspect ratio, XPress creates a new file based on the multi-layer SVG template. The Multi-layer SVG template simplifies the task of composing a scene using XPress. With this template, you can interactively add, position, scale, skew and rotate several media elements such as video clips, images, texts and animations on the same scene.

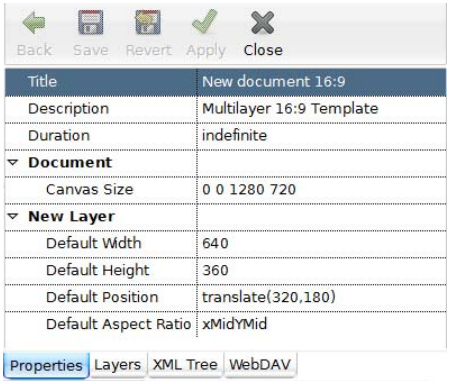


FIG. 17 Modifying the Document Parameters

Composing a Scene

1. Right-click in the main viewing pane and select *New SVG* to create a new_[AR].svg file. You can rename it if you wish. Double-click on the file to open it with XPress.
2. Add new media, or text to your scene, using one of the following options:
   - Drag and drop the media into the *Layers* tab of the *Editing* panel
   - Drag and drop the media into the *Preview* panel. If the cursor doesn't show the + sign, press the CTRL key while dropping the media.
   - Press the icon on top of the *Preview* panel to create a new text layer.
3. Modify the media:
   - Modify the depth position moving the media up or down in the list of media appearing in the *Layers* tab (Use keyboard shortcuts: Crtl-up/Ctrl-down).
   - Use the *Preview* panel to interactively position, scale, skew and rotate the new media (for more information, please refer to the *Interactive Editing* on page 11.)
   - Add entry and exit effects to your media.
4. Save your document.

Customizing a Multi-Layer SVG Document

For each new multi-layer SVG document, you can configure the following properties:

- **Title:** an informative title for the document
- **Description:** an informative description of the document
- **Duration:** the duration of the document. If set to indefinite the document will not loop. If set to media, the duration of the document will be computed automatically and the document will loop at the end of the playing time. It is also possible to set an explicit duration to a given number of seconds.
- **Document:** group for the document properties.
- **Canvas Size:** the size of the document in the following syntax:

  origin x, origin y, width, height.

  The ratio between the width (3rd number) and the height (4th number) determines the aspect ratio of the document.

The first 2 numbers describe the origin of the document and should be left at 0.

- **New Layer:** group for the default properties of all new layers
- **Default Width:** the width of a new layer. By default, the new layer occupies the whole document and thus this has the same width of the Canvas.
- **Default Height:** the height of a new layer. By default, the new layer occupies the whole document and thus this has the same height of the Canvas.

- **Default Position:** position of the top-left corner of the new layer. By default the new layer is located at the top-left of the screen (0,0).
- **Default Aspect Ratio:** you can choose if you want a new layer to keep its aspect ratio (xMidYMid) or to be scaled (none) when inserted into the document.

NOTE

You can change the above parameters at any time. In particular if you want to add new media to your document with different sizes and default positions, you can modify the document parameters accordingly for each new media that you add.

In addition to the *Properties* tab, the multi-layer document activates a *Layers* tab in the editing panel.

The *Layers* tab allows you to add or remove layers of media into your document. The following operations are possible:

- Add a new media layer by dragging-and-dropping any media to the *Layers* tab of the *Editing* panel. The new media will be linked automatically to your document.
- Add a new text layer using the icon in the *Preview* panel, or using the pop-up menu.
- Delete an existing layer (using the pop-up menu or the delete key)
- Change the order of the layer by using keyboard shortcuts (*Ctrl+up*/*Ctrl+down*), or by manually dragging and dropping the list).
- Position, scale, skew and rotate the new media by interacting with it in the *Preview* panel.

Each layer position and size can be edited on the *Preview* panel as described in the *Interactive Editing* on page 11. Multimedia effects can be added to any media as described in the *Multimedia Effects* on page 23.

Example: Editing

A typical document will be composed of a background image (the first layer), and a combination of media to create the complete scene.

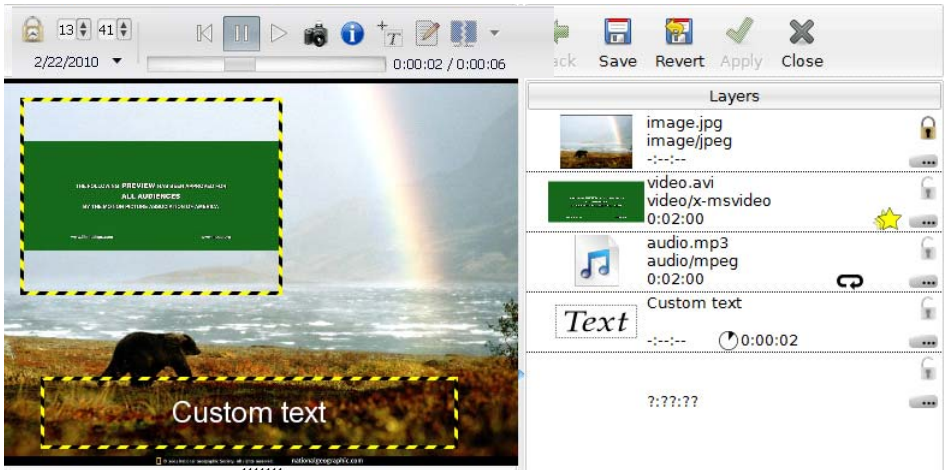


FIG. 18 Layers in a sample document

For instance the following example is composed of 4 layers:

- A background image.
- A video on the left.
- Some text on the bottom.
- Background music. (This layer is not visible, as it contains only audio).

Playlist Template

The user can create a scene using the multi-layer SVG document. To compose an entire presentation, it is necessary to display one scene after another, possibly over several scenes, using a playlist. The simplest way to create a new playlist is by right-clicking the browsing panel and selecting *New Playlist*.

The playlist template allows you to create a temporal sequence of media files: videos, images, texts, multi-layer documents, etc. Each media will be played one after the other in the order determined by the user.

Composing a playlist SVG document

1. Right-click in the browsing area and select *New Playlist* to create a new_playlist.svg file. You can rename it whatever you wish.
2. Open the newly created SVG file. In addition to the *XML Tree* and *WebDAV tabs*, the editing panel will open the *Properties* and the *Playlist* tabs.
3. In the *Properties* tab, you can customize some of settings of the playlist, such as the maximum slide duration, the document size, and the transition between slides.
4. In the *Playlist* tab, you see the default elements of the playlist. These elements contain some instructions on how to use the SVG document with XPress that you can directly read in the *Preview* panel.
5. To compose your playlist, simply drag and drop single files or groups of files (any media file: video, images, SVG, etc.) from the browsing area into the *Playlist* tab. The files will be listed in the *Playlist* tab.
6. Customize your playlist by changing the order of the files in the Playlist tab and by changing the duration of individual media using the "..." icon. Apply your changes to preview them in the Preview area.
7. Save the final playlist.

How to Customize a Playlist SVG Document

The playlists created using the *New Playlist* menu (or using the *My Templates/new_playlist.svg* file) allow the user to select a set of predefined transitions between the slides. This playlist template instructs XPress to expose the following parameters in the *Properties* tab (FIG. 19):

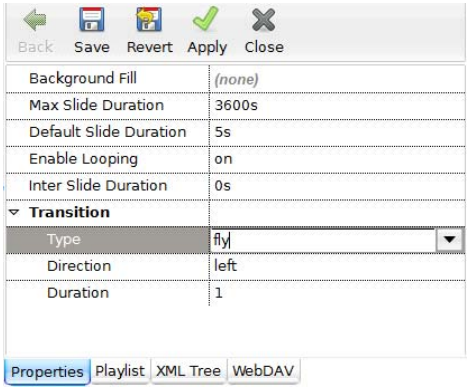


FIG. 19 Properties Tab

- Background Fill: Specifies the color of the background. Can be set to 'none' to indicate a transparent background.
- Max Slide Duration: Specifies the maximum duration of one slide (media). If a media is longer than this duration, it will be truncated. This option is very useful for playlist of images, as images have no duration.
- Default Slide Duration: Specifies the default duration of one slide (media) if the media doesn't have a duration (like images).
- Enable Looping: Specifies whether the playlist should loop or not when reaching the last media file from the list.
- Inter Slide Duration: Specifies the interval between the display of two media files. If set to a value different than zero, the background will be shown during this amount of time at the end of every media.
- Default Transition: Groups the transition settings of the playlist
- Type: Type of transition between the slides. The following transitions can be selected:
  - **none:** No transition between each media. This is equivalent to a scene cut.
  - **fade:** The new slide fades in while the old slide fades out.
  - **fly:** The new slide flies in, while the old slide flies out, according to the direction set by the user.
  - **wipe:** The new slide is wiped on top of the old slide, according to the direction set by the user.
  - **random:** Select a different random transition for each slide.
- Direction: For the fly and wipe transition, select the direction of the effect.
- Duration: The duration of the transition effect.

In addition to the *Properties* tab, playlist SVG documents instruct XPress to open a *Playlist* tab in the editing panel. This tab lets you compose the playlist and the order in which media elements will be played.

The following interactions are possible from the *Playlist* tab:

- **Add a new media.** You may use different ways to add a new media:
  - Drag-and-drop any media to the *Playlist* tab of editing panel. If the new media is dropped on top of an existing media, it will replace it. Otherwise it will be inserted where it was dropped.
  - Click on the "..." at the bottom right of the table. This will open the *Media* dialog, and allows you to type the URL of your media. This is useful if you want to add media from remote sites.
  - Use the pop-up menu and choose *Insert*. This will insert an empty media at the selected position. The empty media can then be edited using the "..." at the bottom right of the line.
- **Delete an existing media**. Use the pop-up menu or the Delete key.
- **Change the order of the media.** Using the pop-up menu, or using the *ctrl+up/down* keys.

It is not only possible to create a playlist of media files, but also a playlist of text boxes if needed. Media and text boxes can also be used in the same playlist.

It is possible to control the duration of each media and the number of times it will be repeated (if needed). The Playlist tables shows for each media:

- The media URL.
- The media type.
- The total playing time of the media. A looping symbol is added to the bottom right of the column if the media will loop at least once (FIG. 20).

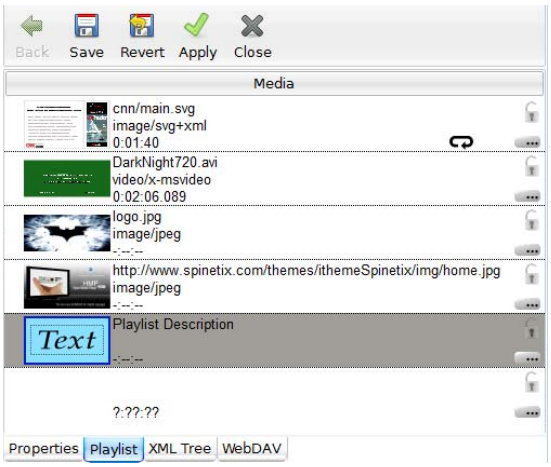


FIG. 20 Playlist Tab

Special News Templates

To simplify the creation of dynamic content, XPress is provided with a set of active templates that can be customized by the user and that perform specific actions. The active templates use JavaScript for dynamic creation of SVG documents with specific graphical layouts and up-to-date content.

In this section, we discuss a number of active templates we have developed to retrieve RSS content. We call these templates *news templates*.

RSS is a family of Web feed formats used to publish frequently updated content such as blog entries, news headlines, and podcasts. RSS documents are XML documents with specific structures and tags which can be retrieved from the Internet or other network locations.

The news templates instruct the IS-SPX Player to extract from remote RSS files the "content" and apply a customizable graphical layout which may include the text positions in the scene, the font properties, the background image and the animation of each element of the scene.

The same template can also use XML, Text, CSV or iCalendar files as data source.

A few examples of templates are suggested that display the text on the screen according to predefined graphical layouts. All the templates can be found under the My Templates/ folder after XPress installation. The templates are split into two categories: The *Layout Templates* and the *Format Helpers*.

Layout Templates	
Template Name	**Description**
news.svg	Display news from RSS, XML, Text, CSV or iCalendar in a formatted manner. Supports single or multiple items per page.
news_ticker.svg	Display news text in a scrolling banner (left/right or top/bottom).
news_crawler.svg	Display formatted news using scrolling (left/right or top/bottom). Can be used to display logo/ media in a ticker or multi-line tickers.

Format Helpers	
Helper Name	**Description**
bouncing_text.svg	Text too large to be displayed will 'bounce' left to right or top to bottom
fit_text.svg	Text size will adapt to the area available.
date_time.svg	Display a date/time in a formatted manner.
switch.svg	Display different media depending on the item content.

Composing a Live News Document

To compose a live news document:

1. Copy the news template SVG file from *My Templates/News* into your project.
2. Open a new exploring tab of your current project in the browsing panel.
3. Open an additional exploring tab located in *My Templates/News*.
4. Drag and drop the template file you have chosen from the news tab into your current project tab.
5. Open the newly created SVG file. Select your news source using the RSS Feed Source entry. Type the URL of the RSS news to use an external news feed.
6. Customize the display of your news source:
   - Press the **Edit** button.
   - Move or resize the text/media box in the preview panel as desired.
   - Edit the text box content. The [SPX][title] and [SPX][description] will be replaced by the content of the RSS news.
   - Add news text or media boxes using the *Layer* tab.

7. Customize the settings of your news source using the *Properties* tab. The *Feed Type* lets you select the type of your input data. The *Feed Update Policy* lets you control when the feeds should be updated. Additional settings can be found depending on the templates.
8. Save the final document.

Customizing a News SVG Document

All the templates SVG files provided in My Templates/News let you customize their layout using a different set of parameters:

news.svg

This template should be used when you want to display one or more news feeds on a single page. Each new feeds is shown one after the other, like a slide show. The following features can be configured:

- Supports entry and exit effects to add movement to the displayed news feed.
- Supports one or multiple news items per page.

You can control:

- **Item Duration:** The duration of each pages.
- **Document:** The size of the document.
- **Multiple Items Per Page:** The number of items per page and the location of each new item. You can also add an offset between the display of the news feeds on the same page.

news_ticker.svg

This template should be used when you want to display a scrolling text coming from an external source. The size of the scrolling text will automatically adapt to the size of the document. The following features can be configured:

- **Layout:** Layout of the text. Use [SPX][title] and [SPX][description] to display respectively the title and the description.
- **Direction:** Support horizontal and vertical scrolling.
- **Crawling speed:** Crawling speed expressed as a percentage of the height of the characters.
- **Spacing:** Spacing between two news items.
- **Background:** Background color and opacity.
- **Advanced:** Number of lines of text shown simultaneously (in case of vertical scrolling).
- **Baseline:** Vertical position of the characters. Increase this value if no characters are shown.

news_crawler.svg

This template should be used when you want to display scrolling text and a logo, or when you want to have more control over the layout of the displayed news. This template combines the features of the *news.svg* and the *news_ticker.svg* templates.

Effects cannot be used in this template.

Using Format Helpers

The *Format Helpers* are a set of files which can be added as a multimedia layer in the *news.svg* and *news_crawler.svg* file to simplify the construction of the layout.

To use the bouncing text or the fit text helper:

1. Drag and drop it on your document like any multimedia file.
2. Open it
3. Change the Text to [SPX][title] or [SPX][description] to display respectively the title or the description of your news.
4. Press **Apply** or **Save** to see the result.

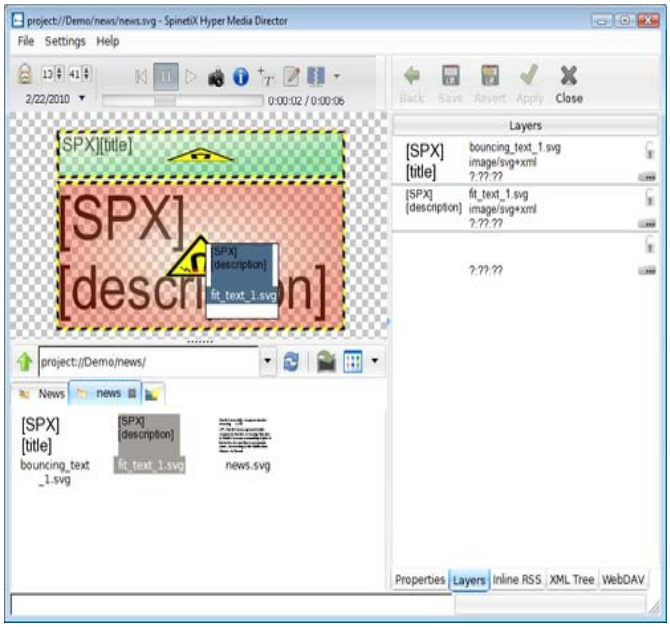


FIG. 21 Adding a Format Helper

bouncing_text.svg

If the text is too large to be displayed in the text box area, then the text will bounce from left to right (or top to bottom, depending on the configuration). When using top to bottom bouncing, you can configure the number of displayed lines.

fit_text.svg

The text size will be automatically adapted to the text box area, ensuring that all the text is always displayed.

date_time.svg

This helper can be used to display any date or time in a formatted manner. You can configure the 'source' for the date using the *Date source* entry (*pubDate* for the publication date of the RSS news for instance). The helper uses the PHP syntax for controlling the date format. You can also use any language for printing the date by filling up the entry under the *Names* group. By default, English, French, Spanish, Italian and German examples are provided in the template.

switch.svg

This helper lets you configure different media to be displayed depending on the value of a news item. This can be used to display an up/down arrow for stock quotes for instance.

Scalable Content and Aspect Ratio of Your Display

The IS-SPX Player natively supports the SVG graphic language. This language is scalable: that means that the graphics are rendered on the fly according to the resolution supported by the display. This guarantees a perfect rendering for texts and graphics, regardless of the resolution of the display. (Note that the scalable property does not apply on images or video content encoded with other formats and included in a SVG document.)

When authoring scalable content, the user should pay attention to the aspect ratio (the ratio between the horizontal and vertical size) of the display. To avoid aspect ratio distortions or black borders, the user should author the content with the same aspect ratio as the target display.

Digital displays may offer different aspect ratios: the most common are 4:3 and 16:9. The first are those typically seen in computer monitors: the ratio between the width and height is in 4:3 proportions. The second is common in recent TV sets where the horizontal size is almost the double of the vertical one (landscape).

To simplify the authoring with the right aspect ratio, XPress comes with two sets of templates: one created for 4:3 aspect ratios and one for 16:9. We have also created intermediate aspect ratios such as 16:10 which are less common. When possible choose the aspect ratio corresponding to your displays.

Example of 16:9 content:

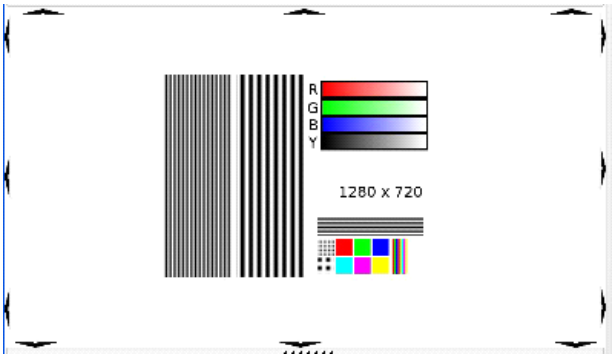


FIG. 22 Example of 16:9 content

Example of 4:3 content:

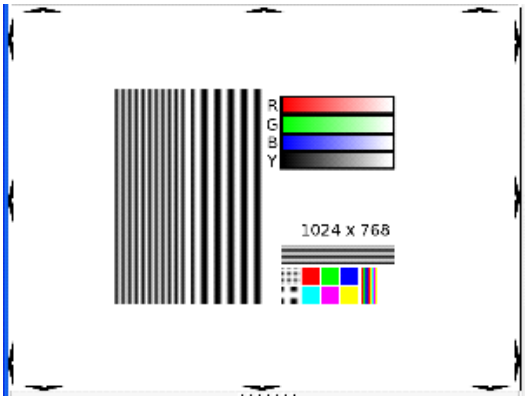


FIG. 23 Example of 4:3 content

It may happen that the same content will be displayed on monitors with different aspect ratios. In this case the following will happen:

- If you use 16:9 templates, the content will be displayed using black borders on the top and bottom when shown on a 4:3 screen.

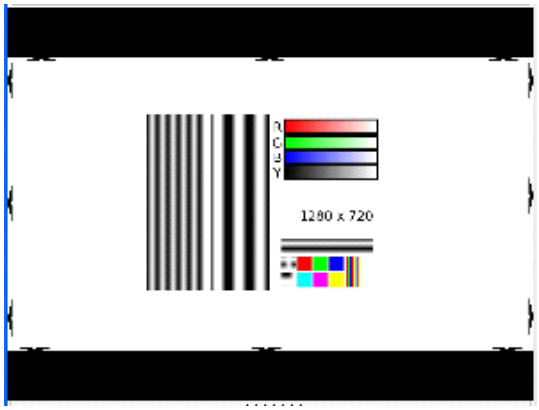


FIG. 24 16:9 templates as shows on a 4:3 screen

- If you use 4:3 templates, the content will be displayed using black borders on the left and right when shown on a 16:9 screen.

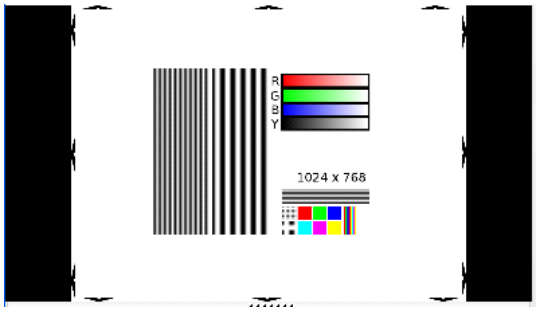


FIG. 25 4:3 templates as shown on a 16:9 screen

- If you use "fit" templates, the content will be scaled to fit the whole screen, regardless of its aspect ratio. This will introduce distortions that may be noticed according to the nature of the content.

  Example of content with a real aspect ratio of 16:9 when fitted to a 4:3 display:

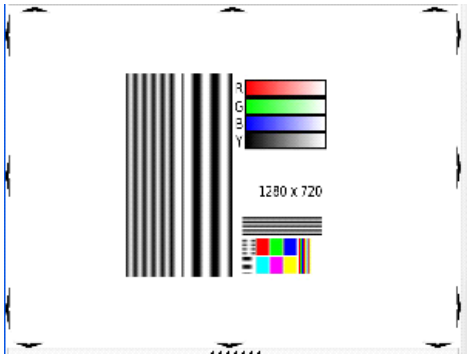


FIG. 26 Real Aspect 16:9 content fitted to a 4:3 display

Example of content with a real aspect ratio of 4:3 when fitted to a 16:9 display

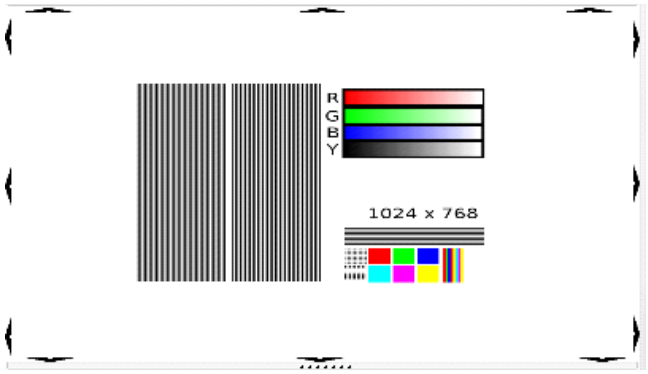


FIG. 27 Real Aspect 4:3 content fitted to a 4:3 display

Note that in XPress, you may change the aspect ratio of the preview panel by sliding its horizontal and vertical dimensions. When previewing SVG content, XPress will do exactly as the IS-SPX Player will do on real displays with the same aspect ratios.

Multimedia Effects

Adding Effects

The XPress software installation comes with a number of default media transition effects files. These may be found in the *Effects* folder under *My Templates*. These effects files can be applied to media elements such as images, video or animations. Effects will animate how the media will appear in the scene (entry effects) and eventually how it will disappear (exit effects). XPress offers an intuitive user interface to apply effects to media elements:

- Open a document that contains a media element such as an image. The preview panel will display the entire scene.
- Drag and drop one of the effects files directly into the preview panel on the media element you want to animate.

In the case of overlap in the media you want to animate, leading to difficulty in dragging-and-dropping media onto the *Preview* pane, then drop the effect file directly on the media property in the *Editing* panel.

Once an effect has been added to a media, it can be configured using the *Media Editing* dialog. The editing panel shows a star icon next to the "…" of the media property to indicate that effects have been added to the media.

To remove the effect, use the popup menu on the media element, and select *Remove Effect* ….

Two kinds of effects are available:

- **Entry effects:** these effects are triggered by the beginning of the multimedia file. These are also know as input or entry effects.
- **Exit effects:** these effects are triggered by the end of the multimedia file. These are also know as output or exit effects.

These transition effects are based on SVG syntax and may be customized. This makes it easy to design a new set of effects using the SVG animation syntax.

Configuring Effects

Once effects have been added to a media file, the *Media Description* dialog will appear (FIG. 28).

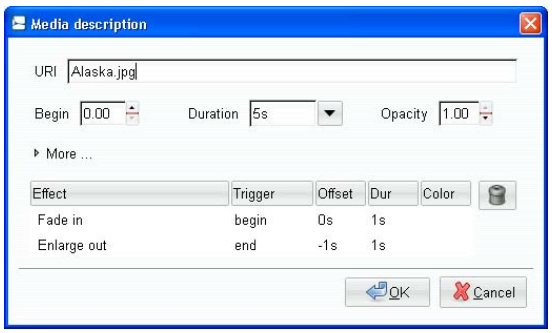


FIG. 28 Media Description dialog

The *Effect* list allows you to remove or to customize effects. It is possible to modify the following parameters:

- **Trigger:** Select if the effect starts at the beginning or at the end of the media file. This parameter is automatically set by XPress when dropping an effect file unto a media element.
- **Offset:** The offset between the beginning/end of the media and the beginning of the effect. For exit effects, the offset is generally equal to the negative value of the effect duration, so that the effect finishes with the media.
- **Dur:** The duration of the effect. This parameter let the user control the speed of the effect.
- **Color:** Some effects have a configurable color, which can be controlled using this parameter.

Selecting an effect and pressing the trashcan icon will remove the effect.

Previewing

From the *My Projects* folder, you can access all the projects that are currently defined. To explore and preview a specific XPress project, simply double click its icon. This will open a new view in the browsing area of XPress. The name of the view is the name of the project.

You can preview the selected XPress project directly with XPress by opening the *index.svg* file. The content will start playing in the *Preview* area of XPress. You can pause the presentation at any time and interactively navigate with a time ladder.

In addition to the *index.svg* file, other folders may be found that contain media content, scripts, etc... The index file is used to concatenate all these media elements and to schedule when they will be played. You can also individually open the other media elements. The preview and editing panel will change according to the document you have opened.

In this example, the project *Demo720p* has been opened in the browsing panel and XPress has created a new view tab: *Demo720p* (FIG. 29).

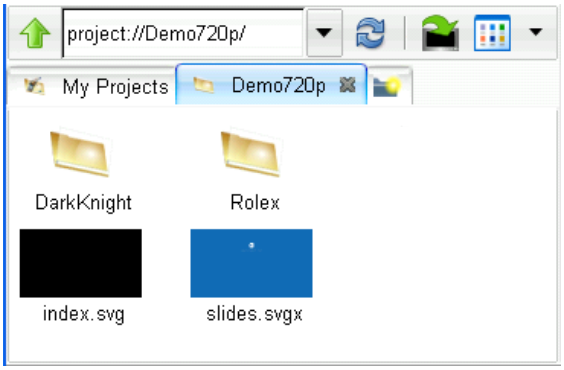


FIG. 29 Demo720p Folder

Preview XPress Content

The XPress content previewed in XPress will look exactly the same as it will be played on the IS-SPX Player. You can use the preview feature of XPress to check the look of your project before it is played on the device.

There is however one difference between images previewed through XPress and those previewed through the IS-SPX Player. Backgrounds that are transparent will be represented by a checkerboard on XPress (FIG. 30), whereas they will be displayed as black on the IS-SPX Player (FIG. 31).

FIG. 30 Transparent background as shown in XPress

FIG. 31 Transparent background as shown on the IS-SPX Player

Publishing

To publish a project manually to a device, do the following:

1. Open *My Devices* in a first tab. Identify the IS-SPX Player on which you want to publish. For more information, please refer to the *Monitoring* on page 26.
2. Open *My Projects* in a second tab.
3. Drag and drop the project you want to copy to the device you want to update.

The above procedure will copy the entire project into the targeted IS-SPX Player. All the modified files will be updated with the new version, all the new files will be added, and all the files already on the IS-SPX Player but no longer in the project will be deleted.

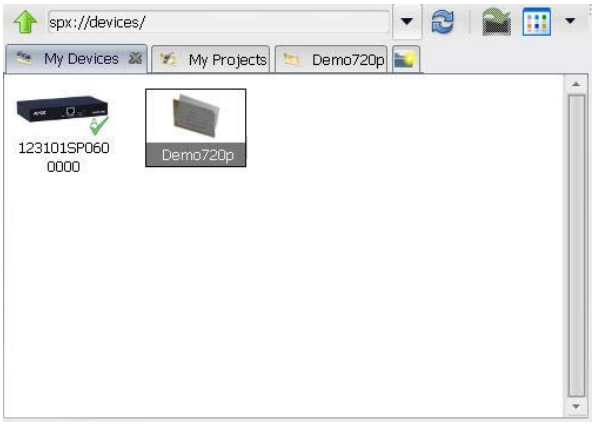


FIG. 32 Copying the project into the IS-SPX Player

NOTE

The publishing might take some time, depending on the size of your project and the speed of the network. Once the project has been published, do not unplug the device without a proper shutdown, or part of the content might not be properly saved on the IS-SPX Player. For safe shutdown of the device, use the http configuration interface.

Using Quick Publish

Once a project has been published manually to the device, it can be published again using the **Quick Publish** button. The **Quick Publish** button is located on the address bar (FIG. 33) and is activated whenever a project is open.

FIG. 33 The Quick Publish button on the address bar

To use Quick Publish:

1. Open your project.
2. Press the **Quick Publish** button.
3. All modifications done on your project will be copied onto the IS-SPX Player.

Monitoring

XPress keeps a list of the devices available in the network using the *My Devices* folder in the workspace view.

The *My Device* view allows you to:

- Monitor the devices connected to your network.
- Automatically scan devices connected to your network.
- Provide details on the status of each device.
- Open a device to browse its content.
- Manually add a device for monitoring.

It is possible to arrange your devices into folders (collections) in order to group devices according to your criteria. It is also possible to rename devices to identify them in a more human readable manner. By default, XPress used the serial number as the name of all devices.

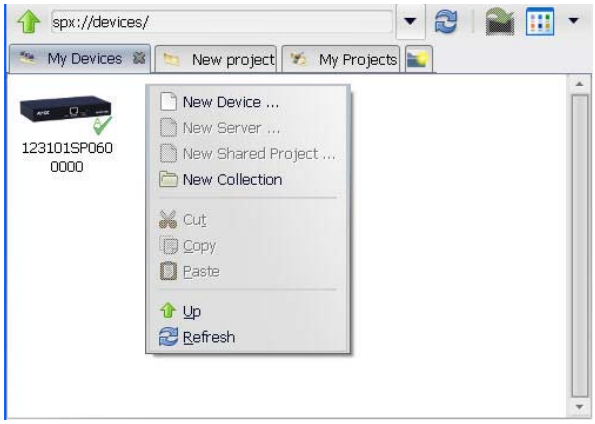


FIG. 34 My Devices tab

Manually Add Devices to XPress

To manually add a device to the My Devices tab, do the following:

1. Open the *My Devices* tab.
2. Right click on the *Workspace* panel and select *New Device...*

FIG. 35 New Device dialog

3. The *New Device* dialog (FIG. 35) appears and asks you to enter a number of details concerning the new device you wish to add:

   Serial Number: If you do not have the serial number of the player you want to add, you can enter here another name here. However, the player you add will not be automatically recognized by the *Discover devices* command as being already defined. You may have two device icons pointing to the same player.

   IP address: Enter the IP address (or the host name) of the device you want to add. This information is necessary to add the device.

   Type: The Type of the device. Currently, only IS-SPX Players are supported by XPress.

Security: You can also directly add user names and passwords for protected devices. If you do not enter a password in this dialog, XPress will ask you to authenticate yourself when trying to publish content to a protected device.

4. The new device is added to the current folder.
5. Once added, you may change the name of the new device from the browsing panel at any time.

Check the Status of the Devices

XPress checks the status of the devices available in *My Devices* each time the refresh button is pressed. If the content server is reachable, it displays a green check on top of the device icon. If the content server cannot be reached, a red cross is displayed instead.

FIG. 36 Device availability

This check mechanism does not inform you on the status of the player. It only states whether or not the player can be reached. To get more feedback about an IS-SPX Player, you should use the *IS-SPX Player Web-Based Configuration Page navigation bar* section in the IS-SPX-1000 & IS-SPX-1300 Operation Reference Guide, available at **www.amx.com**.

Right-click on an IS-SPX Player and select the *Manage...* command from the pop-up menu. This will directly open your preferred web browser on the HTTP server of the selected device.

For the proper management of the status of multiple devices, a SNMP client should be used. The IS-SPX Player is given a product OID of 1.3.6.1.4.1.8942.1.1

Open a Device

To view the content server of a device (access to all the files and folders that have been published on the device), double-click on a device. This will open the content server in the browsing panel and the property page of the device in the editing panel.

The property view shows you information about your device. Do not modify this information, as this might prevent XPress from communicating properly with the device.

By pressing on the *Preview* icon (the camera) it is possible to get a snapshot of what is currently being displayed by the device. The preview is automatically updated every 30 seconds.

You can use the content server to upload content to the device. You can also directly edit the content on the device using XPress. However, it is recommended to use projects to edit the content and then upload the project to the device.

Scheduling

XPress offers an intuitive interface that allows the user to associate to any type of XPress content a schedule in the iCalendar format. Using this feature, the user can plan in advance when the content shall be displayed on a specific IS-SPX Player.

The scheduling interface implemented by XPress is very similar to the one implemented by electronic agendas such as that in Microsoft Outlook: each event you add on the agenda (like when you add a meeting in your Outlook) is associated to a starting date/time and a duration (or an end date/time). In addition, it is possible to associate to an event recurring rules so as to schedule a media to be displayed twice per week on Monday and Friday between 9:00AM and 10:00AM.

Once the planning of the day, week, month or year is finalized, XPress exports the results into a standard iCalendar file. This schedule file has the extension .ics and can be used as any other media to compose your content. The flexibility obtained with such an approach allows the user to schedule full screen content, but also specific elements in the scene (such as different backgrounds, logos, source of information, etc.)

Scheduling the Content Displayed by an IS-SPX Player

When you have created or imported a XPress project (for more information, please refer to the *XPress Projects* on page 4) you have a master file *index.svg* in your project. The following steps explain how you can define when the project will be displayed once you have published it to a specific IS-SPX Player.

1. Create a new schedule document in your project. Simply right-click the browsing area in your project and select New Schedule from the pop-up menu. A new.ics file will be created in your project.
2. Open the new.*ics* file. XPress will display a calendar view in the *Editing* panel, where you can select the current day, week, or month.

3. Drag and drop from the browsing panel of your XPress project any individual media you want to schedule into the calendar view in the Editing panel. Define the start duration of each media and its recurrence pattern, if desired.
4. Repeat the previous step for each media you want to schedule and save your changes.
5. Open the *index.svg* file of your XPress project. Drag and drop the new.*ics* file as a new layer into your *index.svg* file.
6. Save your changes.

The file *index.svg* may already contain layers. You may decide to remove the existing layers and leave a single one associated to the new.*ics*. In this case when you publish your project to a IS-SPX Player, the content will be played according to your schedule. You may also decide to combine your schedule with a background image, which would be an additional layer in your *index.svg*. This choice would guarantee that at least the background image will be displayed for those time periods when you have not planned any content. Multiple layers may also be combined with multiple schedules.

Create a New Schedule

To create a new schedule in XPress:

1. Open the project where the schedule needs to be added.
2. Use the right-click pop-up menu to create a New Schedule (FIG. 37).
3. A new file *new.ics* will be created. This is the scheduler.
4. Rename the scheduler to an appropriate name for your project.

The scheduler can be dropped into a multilayer document or a playlist as any other media.

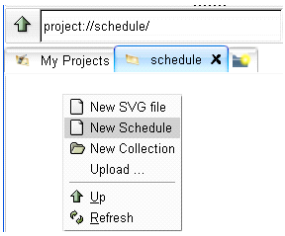


FIG. 37 Schedule menu

Open a Schedule

Double-click the schedule document (*.ics) to open the *Calendar view* in the *Editing* panel.

NOTE

The Calendar view uses the local time settings of your PC.

The *Calendar* view is organized into 4 zones (FIG. 36). Each zone has been highlighted with a different color.

- **Management options:** Allow you to change the time frame that will be shown in the Editing zone. You can choose between day, week, month and year, or create a new empty event.
- **Calendar:** Allows you to select which day will be visible in the Editing zone.
- **Event Options:** This allows you to control the start and duration of an event and eventually its recurrence.
- **Editing zone:** This allows you to view and edit the events in your schedule. Different time scales can be selected from the *Management* zone. Use the day and week scales to change positions and the durations of the events. Use the month and year view to have an overview of your schedule.

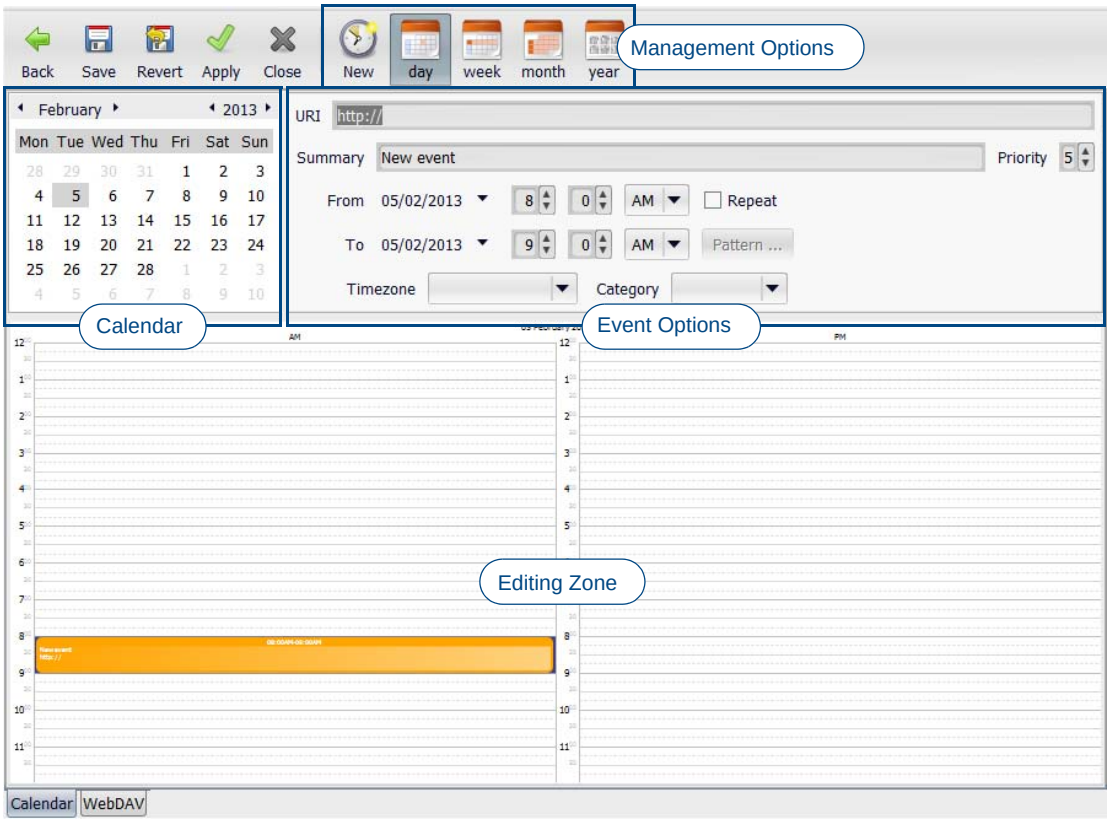


FIG. 38 Calendar view

Add Events To a Schedule

To add an event to a schedule:

1. Open your schedule
2. Select the day or week view from the Management zone.
3. Drag and drop the media you want to schedule into the editing zone of your calendar.
4. A new event will be added where the media was dropped.
5. You can change the start and the duration of the event by enlarging the event and moving its position in the calendar:
   - Clicking on the event allows you to move it.
   - Clicking on the top or bottom of your event allows you to change its duration.
6. The Event zone lets you fine-tune the recurrence properties of your event if needed.
   - Any type of media (animation, video, or image) may be used in a calendar.
   - If you select multiple files in the browsing panel, then all the files will be added one after the other in the calendar according to their default duration (FIG. 39).

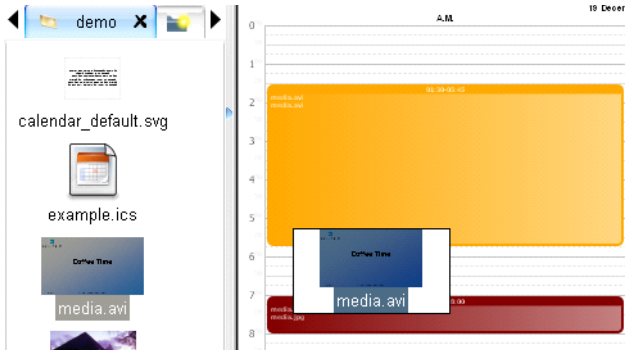


FIG. 39 Adding an event to the schedule

Modify the Scheduling Properties of an Event

You can edit the properties of an event using the Event zone (FIG. 40).

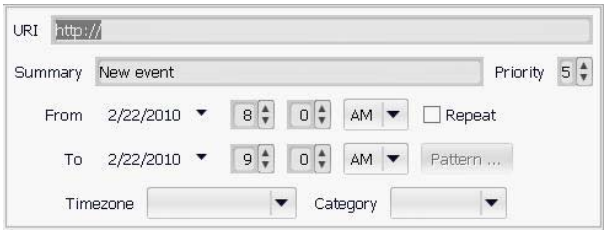


FIG. 40 Scheduling properties of an event

The following properties can be edited:

- URL of the event: Link to the scheduled media. This is automatically modified when you drop any media on the event.
- Summary: Title for this event. By default, it is the same as the filename of the media, but it can be modified to any suitable name.
- From, To: The start and end date of the event. This can be used to fine-tune the duration of the event.
- Priority: In case two events occur at the same time, only the event with the higher priority will be displayed.
- Repeat and Pattern: Allows you to specify repeating patterns for your media.
- Time zone: Allows you to choose the time zone for your event.
- Category: Used for improving the visual interface and choosing colors for the different events. This has no impact on the displayed content.

Define Default Events and Recurrences

Define a Default Animation

The scheduler allows you to specify a default animation to be displayed when no events have been scheduled to a given date/hour. If no events have been specified, the scheduler tries to open the SVG file *calendar_default.svg*. If the file is available, then it will be displayed.

To create and manage *calendar_default.svg*:

1. Create a new document using the right-click pop-up menu on the preview panel.
2. Rename it to “calendar_default.svg”.
3. Open it.
4. Drag and drop you default media (video, animation or image) into the preview panel.

Define Recurring Events

Events can be scheduled as a single-time event or as a series of repeating events. By default, all new events are single-time events. To transform an event into a repeating event, you need to check the *Repeat* check box in the *Event* zone. By default, the event will repeat every day indefinitely.

NOTE

*Depending on the version of XPress, it might be necessary to click the **Pattern** button and then the **OK** button to confirm this repetition.*

To change the rule for a repeating event, press the **Pattern** button, which opens the recurrence rule dialog (FIG. 41). This dialog allows you to create fairly complicated recurrence rules.

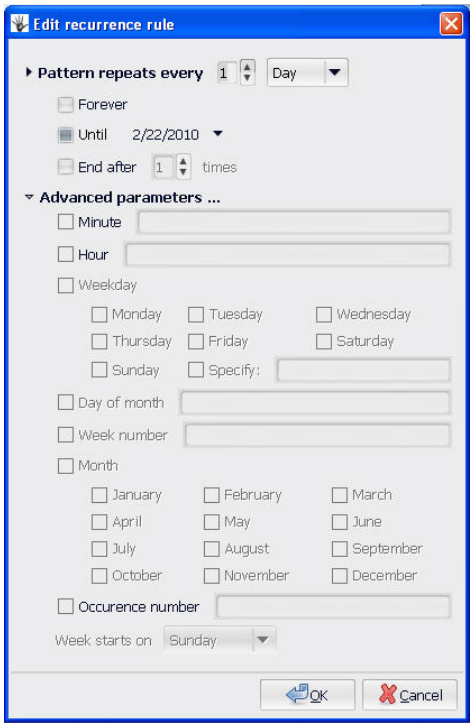


FIG. 41 Edit recurrence rule

Creating a basic recurrence (such as every day, every 2 hours or every week) is fairly straightforward. For instance:

- For an event repeating every day, select “Pattern repeat every 1 Day”. You have the option of selecting a number of repetitions or the date of the last repetition.
- For an event repeating every 2 hours, select “Pattern repeat every 2 Hour”.
- For an event repeating every Monday, select “Pattern repeat every 1 Week”. Make sure your initial event was occurring on a Monday.
- For an event repeating the first day of every month, select “Pattern repeat every 1 Month”. Make sure your initial event was occurring the first day of the current month.
- For more complex rules, you need to use the second part of the dialog. For instance:

  For an event repeating every week day, select “Pattern repeat every 1 Week”, and check “Weekday is”. Then check all the days from Monday to Friday.

  For an event repeating at 10h and 15h every day, select “Pattern repeat every 1 Day”, and select “Hour is 10,15”.

Preview Scheduled Content

XPress allows the user to preview scheduled content at a specific date or time. This is useful to validate in advance what will be displayed in the future.

To preview scheduled content at a specific date or time, use the control on the top of the preview panel (FIG. 42).

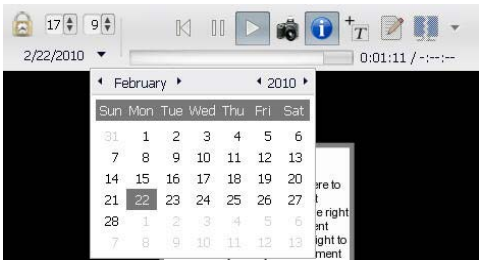


FIG. 42 Preview scheduled content

It is possible to change the time and the date being displayed in XPress using either the clock or the calendar control.

- When the date is modified, the locked-clock icon is highlighted, indicating that the document currently shown in the preview area is not using the local time.
- If you open another document, XPress will open it as if the local date and time is the one you have set. Click the locked-time icon to change the time back to the real local hour.

Data Polling Rates

When configuring a signage project that polls external server feeds, being aware of sample rates is extremely important. Content providers can and will throttle, temporarily block, or even permanently block client requests if their servers are flooded with requests. This is extremely easy to do with a large-scale project. For example, one player requesting an update on a stock source every ten seconds is probably safe. However, since most servers don't update their data that often, best practice is to send as few requests as necessary to satisfy client needs. A good starting point would be twenty or thirty minutes between updates.

One request every ten seconds may not create a red flag for the provider if the requests come from one player, but consider the project increasing in scale to one hundred players, all running the same project. This now leads to one hundred players all requesting an update every ten seconds. Even if these were evenly dispersed, this still leads to ten to one hundred players per second requesting an update. This is likely to be viewed by the content provider or local IT staff as a denial of service attack and they would be justified to take protective action if satisfying IS-SPX Player requests is putting undue strain on their resources. As an alternative, consider increasing the interval between update requests, or introduce an aggregate server on your own network to serve up the data retrieved from the original provider. The aggregate would be one request every ten seconds to the original content provider and it would handle the subsequent requests from your own network. RSS feed sources will often include a max-age set in the XML to advise how often the content cache should request an update from the server.

It should also be noted that if an aggregation server is employed, it is the user's responsibility to ensure that such an aggregation of that data matches the style and format of any given data request precisely.

Remote Control

Overview

The IS-SPX Player has powerful remote control capabilities. You can use the serial port or the IP connection to control the content source, control the playback of animations, video, or audio, and display messages or contextual animations on an external event.

Three main methods can be used to control the IS-SPX Player:

- Network control. This may be done from a PC, a URL to the this player or from another IS-SPX Player.
- **The serial port of the IS-SPX Player.** A dedicated protocol file must be written to translate RS232 commands to a series of actions to be performed by the IS-SPX Player.
- **The USB port of the IS-SPX Player.** Human Interface Devices (HID), such as a keyboard and mouse, can be used to control the IS-SPX Player.

Remotely controlling the IS-SPX Player lets you change the content displayed on the screen or pause a video based on the user input.

This guide covers the multiple aspects of remotely controlling the IS-SPX Player, using the templates that come with the installer and those available at **www.amx.com**. We'll also see how to write your own JavaScript code for more advanced uses.

All examples follow the same structure:

- For IP control, the commands will be sent over TCP with the telnet utility from the command line or from sample client-side scripts in HTML pages.
- The serial port examples assume a simple protocol where each command is a single character, so it can be exercised easily by connecting a PC to the serial port and sending the right keys.

Remote Control of Content

The simplest solution to write content that can be switched remotely is to use the *menu_media.svg* template in the *Interactivity* folder in the *My Templates* Workspace pane. The *menu_media.svg* file lets you control which media is currently being displayed, depending on the value of a shared variable. (For more information on shared variables, refer to *Shared Variables* on page 85.) This same template, used together with the *menu_buttons.svg*, can be used to create menus on the IS-SPX Player.

When To Use Remote Control

The menu_media.svg template offers a menu blueprint that can:

- Replace the current content temporarily for a predetermined time on an external event.
- Replace the current content indefinitely on an external event.
- Replace only part of the content temporarily or indefinitely, using the menu media as one of the layers in a multilayer document.
- Replace the content on external events, with the insertion of a short transition sequence dependent on the trigger event.

How To Use Remote Control

The first step is to define a table of all the possible media contents and associate a page ID to each one of them (FIG. 43).

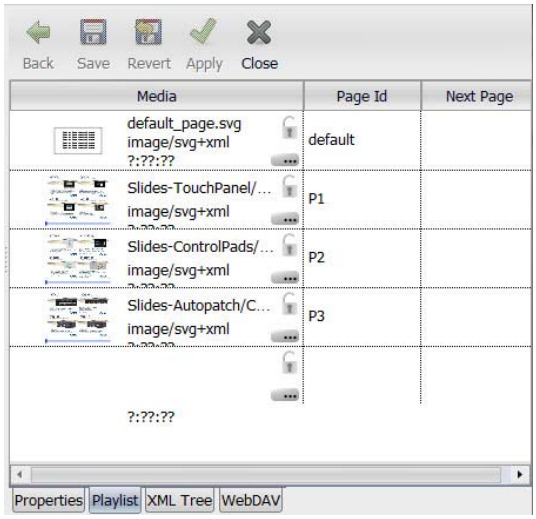


FIG. 43 Page ID table

Next, associate the menu document with a variable name. This configuration is done in the *Properties* section of the *XML Tree* tab of the template (FIG. 44).

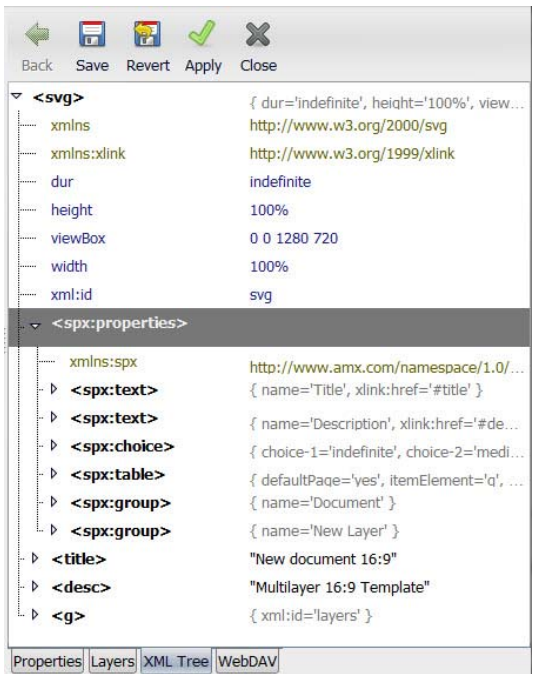


FIG. 44 Properties section of the XML Tree tab

Whenever the content of this variable is changed by a network or serial port command, the menu will switch to display the media with a matching ID. By default, the first content of the list is displayed.

Using the *Properties* tab, you can select the default transition when switching content. You can also select the duration of the transition, and the duration of the pause between two items, if needed. (For more information, please refer to the *How to Customize a Playlist SVG Document* on page 16.)

To decide whether to play the new content indefinitely or only for a predetermined time, you can adjust the media duration using the *media property* dialog. If the media has a finite duration, it will switch to another of the media files if you have entered a media ID in the *next page* column. This lets you trigger a small playlist if needed (FIG. 45). If you leave the *next page* field blank, the action taken depends on the *Next page by default* option. You can chose to switch back to the first page (the default), to go to the next page or to play the same content again.

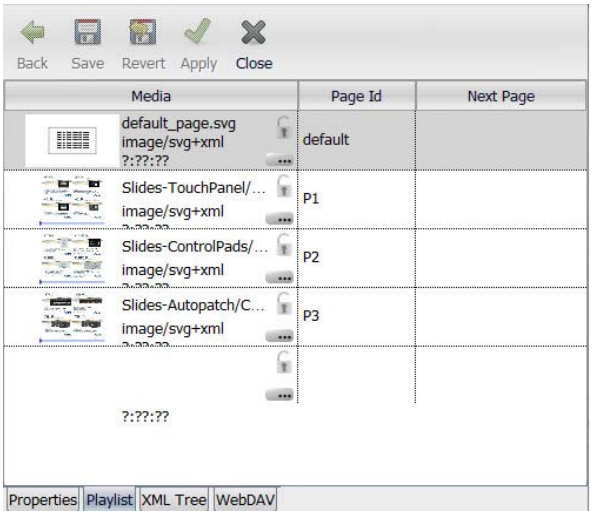


FIG. 45 Playlist

Finally, some values have predefined functions. By default, these are:

- **NEXT**: move to the next page
- **PREVIOUS**: move to the previous page
- **PAUSE**: pause playing interactively
- **PLAY**: resume playing after a pause

You can change the names for those special values in the *Interactive Controls* property group in the *Properties* tab (FIG. 46).

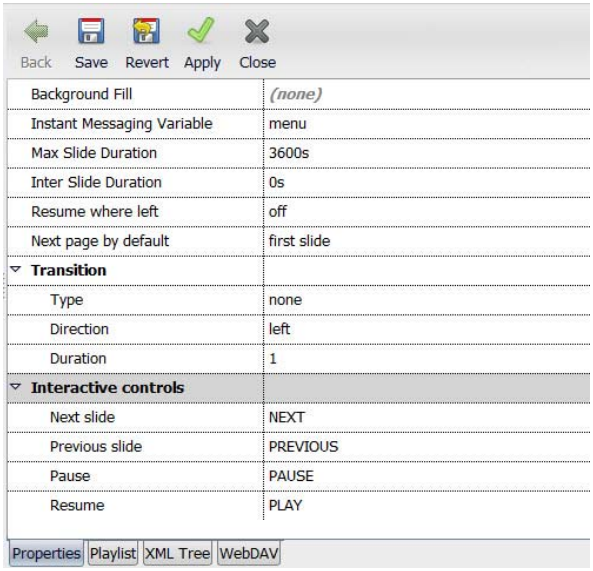


FIG. 46 Interactive Controls property group

Triggering a Content Switch Over IP

Modifying the media is done using shared variables. For more information on shared variables, please refer to *Shared Variables* on page 85.

To change the current media id over IP, you have to send a string like the one below to TCP port 1234 on the IS-SPX Player. The network API must be enabled on the *Network Settings* page for the IS-SPX Player to accept the connection. The port number can be changed on this page as well. (For more information, please refer to *Enabling the Network API* on page 88.)

```
UPDATE "menu" "P1"\r\n
```

If you do not have an IS-SPX Player at hand, you can still test the remote control protocol with content running in the XPress application on your PC. For this, check the *Enable Network API* box in the *Settings* menu (FIG. 47).

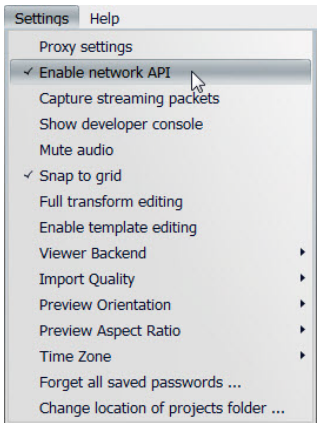


FIG. 47 Settings menu - Enable Network API

NOTE

When enabling the network API via XPress, the TCP port number is also 1234.

This assumes that the control variable is the default name (“menu”) and you want to switch the content to one with the ID equal to “P1”. *\r\n* means you send the characters with ASCII code 13 (CR) and 10 (LF) to terminate the command. This can be tested with the telnet utility, which sends this sequence when you press *Enter*.

The network API also supports setting shared variables via an HTTP GET request. This method does not allow persistent connections, but it is easier to access from a client scripting language supporting the XMLHttpRequest API, such as JavaScript in web browsers.

To generate a command equivalent to the one above, enter the following URL in a web browser:

```
http://[is-spx-1000_address]:[port]/update?menu=P1
```

Example: Testing

To test, publish the menu_media.svg template file to an IS-SPX Player without modifying it. If you have not already done so, enable the network API on the device.

NOTE

For more information on enabling the network API, please refer to the IS-SPX-1000 & IS-SPX-1300 Operation Reference Guide, available at ***www.amx.com****.*

While on the configuration home page (status), note the IP address. Assuming the IP address of the IS-SPX Player is 192.168.1.10, type in: “telnet 192.168.1.10 1234" on the PC, then the switch command, then Ctrl+] and “quit” to terminate the connection. This should look like this:

```
telnet 192.168.1.10 1234
Trying 192.168.1.10...
Connected to spx-1000-001d500000fa.amx.local (192.168.1.10).
Escape character is '^]'.
UPDATE "menu" "P1"
^]
telnet> quit
Connection closed.
```

As soon as you hit enter after the update command, the display should switch for 5 seconds, then revert to the default page.

To test the HTTP GET method with the previous example, enter the following URL in a web browser:

```
http://192.168.1.10:1234/update?menu=P1
```

The GET method can also be embedded in an HTML page. The example below will let you switch the document each time you click on the button.

```
<html>
<script>
function set( name, value ) {
var request=new XMLHttpRequest();
request.open("GET", "http://192.168.1.10:1234/update?"+
encodeURIComponent(name)+"="+encodeURIComponent(value));
request.send();
}
</script>
<body>
<form>
<button onclick="set('menu','P1')">Click</button>
</form>
</body>
</html>
```

Switching Content With the Serial Port

To change the current media via the serial port, you need to do the following:

- Create a protocol file (*.fsm* format) describing the actions to be performed depending on the input on the serial port.
- Upload the protocol file to the IS-SPX Player using the Serial Port page of the HTTP configuration interface.
- Enable the protocol file by selecting it in the protocols drop box.

The protocol file should trigger an <update> action on the variable when matching the input on the serial port.

For instance, the sample below will trigger an update of the menu variable when an '1' or a '2' is received on the serial port.

```
<?xml version="1.0" encoding="utf-8"?>
<protocol target='monitor' startup='start'>
<state xml:id='start'>
<onInput match="1" >
<update name='menu' value='L1' />
</onInput>
<onInput match="2" >
<update name='menu' value='L2' />
</onInput>
</state>
</protocol>
```

Switching Content With a USB HID Device

You can change the current media with a USB keyboard, or any USB device of the HID class that emulates one, including IR remote controls. The *keystroke.svg* template lets you associate keystrokes with a request to update the value of the shared variable that controls the current media.

You should have only one *keystroke.svg* template in each project, preferably as the background layer of the top *index.svg*. If you have multiple layers which accept keyboard events, you have to make sure the correct one gets the focus. This template has no displayable content and is fully transparent.

Specify the mapping from key to value in the *Key Identifiers* tab in the *keystroke.svg* file (FIG. 48). Key identifiers may be augmented with “modifiers”, so as to match on the key stroke only if some other control key is down at the same time. The valid modifiers are *Ctrl*, *Alt*, *Shift*, *AltGr.* and *Meta*. For instance, to match a stroke on “A” while the holding the *Control* key down, the key identifier is “Ctrl+A”.

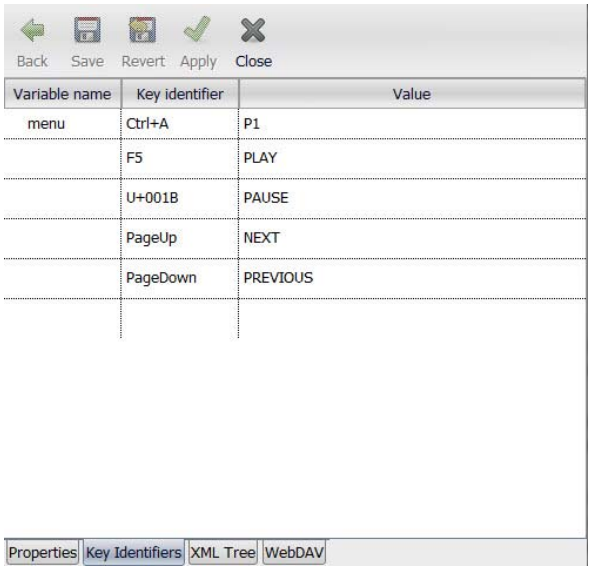


FIG. 48 Key Identifiers tab

In the example (FIG. 49), a mapping is made for the key codes sent by common USB remote controls used for PowerPoint presentations, while *Ctrl+A* trigger the alternate content of the sample.

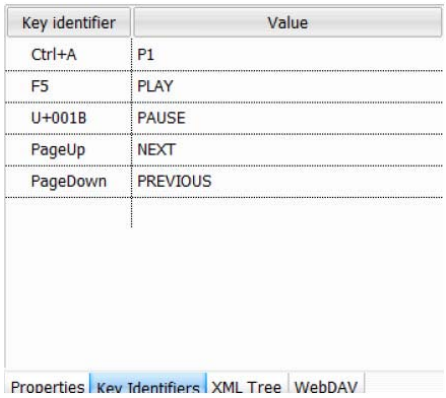


FIG. 49 Key code example

Switching Content From Another Device

It is possible to change a variable remotely from another device simply by creating a shared variable object with the name of the control variable suffixed by the @ sign and the hostname or IP address of the target device, and if not using the default port, followed by a colon and the port number. For instance for the control variable “menu” on the device “target”, the variable name is “menu@target”.

To generate this command you can use the *menu_buttons.svg* template located in the *Interactivity* folder. (For more information on the *Interactivity* folder, please refer to *Interactivity Template Files* on page 63.). This template let you configure 'buttons' and associate each button with a given action. This template can be used to control local or remote shared variables depending on the presence of the @ sign in the variable name.

The example below shows the configuration of the file to control the value of the variable 'menu' located on the remote 192.168.1.10 host (FIG. 50).

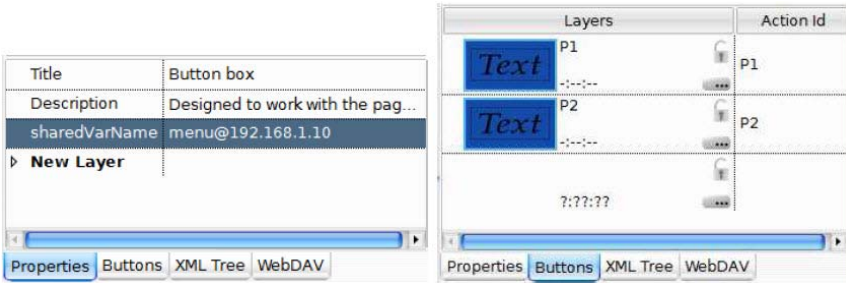


FIG. 50 Controlling the value of the variable

It is also possible to use JavaScript directly to control the value of the shared variables. The sample code below assumes that we want to control the content on 192.168.1.10 by clicking on a rectangle.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" xmlns:ev="http://www.w3.org/2001/
xml-events" viewBox="0 0 1280 720" viewport-fill="white">
<rect x="540" y="260" width="200" height="200" fill="blue">
<handler ev:event="click">
<![CDATA[
var v=createSharedVariable( "menu@192.168.1.10" );
v.set( "P1" );
]]>
</handler>
</rect></svg>
```

Remote Control of Playback

The *playback.svg* template in the *Interactivity* template folder acts as a transparent container for a single media file and lets you control its playback. Using this template, you can pause and restart any type of document including animations, playlists and videos.

When To Use Remote Control of Playback

You can use it to control the main document or for individual media files. It supports the following operations:

- Pausing and resuming playback.
- Changing the current time position.
- Synchronized control of the same media on multiple screens.
- Control from multiple clients over the network and/or serial port.

How To Use Remote Control of Playback

The control is based on a shared variable, the default name of which is “playback”. In its basic usage, it takes two values: “PLAY” or “PAUSE”.

For playback control on multiple screens or better resilience, an optional timestamp can follow the PLAY or PAUSE string (FIG. 51). It is defined as the time position in seconds where playback should pause or resume. This extra parameter enables unambiguous control over the timeline.

If the variable is stored in a central server, this lets you control the playing (or pause) time, even if the target device was off or has restarted in between two commands. It also enables frame accurate synchronization of playback on multiple targets, as well as control over the current time position.

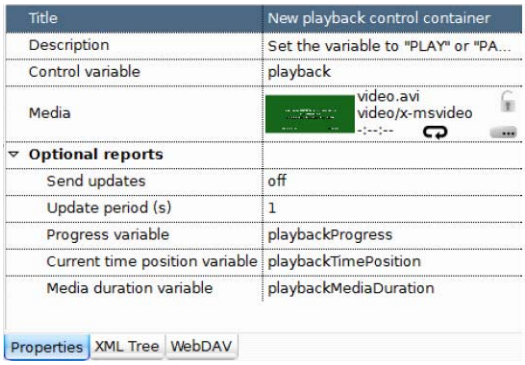


FIG. 51 Playback control properties

The timestamp can also be specified as a relative percentage of the total duration of the media if followed by a percent sign. For instance, to jump to the middle of the media, the value is: *PLAY 50%*.

Interactive Pause/Resume Over TCP

To test the remote control, create a project based on the *playback.svg* template and drop a video media inside, setting it to repeat indefinitely.

To pause the media, send the following command over TCP:

```
UPDATE "playback" "PAUSE"
```

To resume play back, send the following command:

```
UPDATE "playback" "PLAY"
```

Playback will pause, respectively resume, at the time the command is sent. Because of latency, the media may appear to step back in time a fraction of seconds when pausing. This can be avoided entirely by specifying a timestamp in the future for the variable update.

Interactive Pause/Resume Using HTTP GET

You can control the published project directly from a browser using the HTTP GET interface. The following example assumes that the device is using the 192.168.1.10 IP address and the default port (1234) for the network port.

To interactively pause the media, type the following command in your browser:

```
http://192.168.1.10:1234/update?playback=PAUSE
```

To resume playback, send the following command:

```
http://192.168.1.10:1234/update?playback=PLAY
```

GET Embedded In an HTML Page

The playback GET method can also be embedded in an HTML page. The example below will let you play and pause the media from a remote web page.

NOTE

You will need to make sure that the web page is stored on a 'trusted' web site. Otherwise, the security settings of many Web browsers might prevent you from using the XML HttpRequest().

```
<html>
<script>
function set( name, value ) {
var request=new XMLHttpRequest();
request.open("GET", "http://192.168.1.10:1234/update?"+
encodeURIComponent(name)+"="+encodeURIComponent(value));
request.send();
}
</script>
<body>
<form>
<button onclick="set('playback','PLAY')">Play</button>
<button onclick="set('playback','PAUSE')">Pause</button>
</form>
</body>
</html>
```

Interactive Pause/Resume From a Device

Alternatively, you can control the playback of your document using the template *menu_buttons.svg*. The document can be used both on the player of the media file or on remote devices.

To use the *menu_buttons.svg* file to control the playback, you need to configure the name of the shared variable to 'playback' or 'playback@hostname' (FIG. 52). You then need to configure the action related to the button to trigger the PLAY and PAUSE commands.

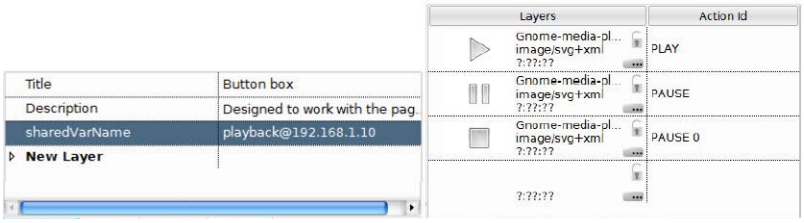


FIG. 52 Configuring the shared variable name

Timed pause/seek/resume

To change the time position of the media, you can change the playback variable at any time and set it to “PLAY” with a time position. For instance, to seek at 6.5 seconds from the start of the media, send the following command:

```
UPDATE "playback" "PLAY 6.5"
```

To pause playback at a precise position, say at 7 seconds, you would send the following command:

```
UPDATE "playback" "PAUSE 7"
```

This is equivalent to a seek to position plus pause operation.

If you wish to resume playback, just send a PLAY command with the same position as the last PAUSE.

It is also possible to use a relative position depending on the content duration to control the pause and play time. In this case, the position should be specified in percentages. For instance, to start play at 10% from the start of the media, send the following command:

```
UPDATE "playback" "PLAY 10%"
```

Getting Feedback On the Playback Progress

It is possible to activate optional reports that will be sent periodically via shared variables. To do so, open the *Optional reports* property group and set the *Send updates* property to "on" (FIG. 53).

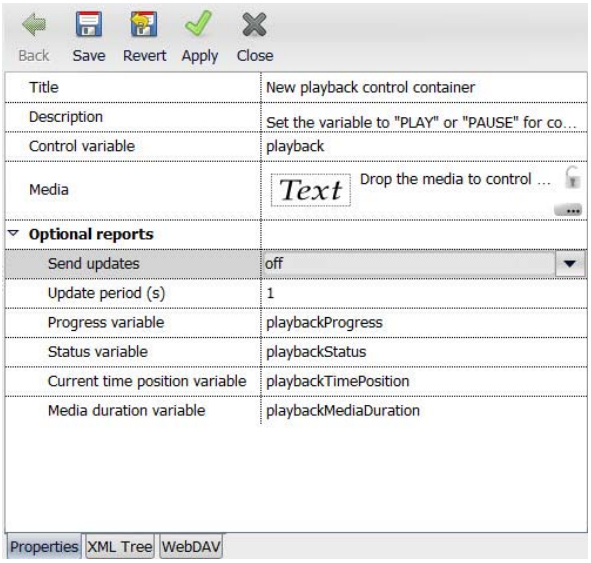


FIG. 53 Optional Reports properties

You can control the periodicity of the updates as well as the name of the variables that receive them. Feedback is sent by default via the following shared variables:

- **playbackProgress**: Relative progress as a percentage of the media duration
- **playbackTimePosition**: Absolute time position
- **playbackMediaDuration**: Reports the duration of the media

For more information on how to access the content of these variable over the network, refer to *Shared Variables* on page 85.

NOTE

If the controlled media is set to loop, the progress percentage resets to zero when the media loops, while the absolute time position does not.

Knowing the Current Position With Extra Accuracy

The IS-SPX Player is precisely synchronized. This feature is used to generate multiscreen content, as it lets you easily infer the time position of the controlled media. The time position is known with better than one-frame precision based on the difference between the time you send the PLAY command and the current time.

NOTE

Make sure that all IS-SPX Players in a network are configured with an NTP time server so they synchronize with UTC.

In order to use the play/pause command with time position, you need to infer the current position as well as when to send the pause command. You can simply copy it from the current value of the playback variable when sending a play command to resume playback.

Remote Triggers

Remote triggers let you display some text, video or animation on the IS-SPX Player controlled from a remote server. For instance printing a message on all the IS-SPX Player of a building or a “GOAL!” message on the screen of the stadium. The same procedure can be used to control which image of video to display from a remote server. There are a choice of techniques to remotely trigger the display of a message, in the form of text, animation, image or video.

Triggering a Text or Image Overlay

The best way to trigger the temporary display of a text or image overlay is to use the *news.svg* template in the *News* directory. In order to configure the new template for this use, follow the steps below:

1. Select a name of a shared variable which you'll use to set the text message.
2. Enter this name in the *RSS Feed Source* property. Append *@localhost* to the variable name so it is recognized as a variable.
3. Select *TXT* for the *Feed Type* property.
4. Set how long you want the message to display in the *Item Duration* property.
5. Set the *Looping* property to *off*.
6. Switch to *edit* mode and draw a single text area with *[SPX][title]* as the content at the location where you want the message to appear.
7. Drop a suitable transition, such as a fade-in on the text area (optional).
8. Insert the template as a new layer in the main document.

The template will remain transparent until you set some text for the chosen variable. Whenever you update the content of this variable, your message will show for the configured duration.

You can follow the same method to display an image, video or animation instead of text. If the image is known in advance you can ignore the content of the variable, just using it as a trigger. To trigger a fixed video or image based on a trigger:

1. Use the news template together with a shared variable as in the previous example.
2. Switch to the *edit* mode and create your layout using your videos or images. Note that as you are not interested in the content of the shared variable, you don't need to use *[SPX][title]* tag in your document. If you plan to play a video, make sure that the duration of the slide is longer than the duration of the video.
3. The videos or images will be displayed for the duration of the slide each time the trigger is received.

Alternatively, you can use the content of the variable as a filename or URL for the image:

1. Use the news template together with a shared variable as for the previous example.
2. Go to the *Layers* tab and insert a new media.
3. Open the *Media Properties* dialog (press the *[...]*).
4. In the *URI* property, type *[SPX][title]*. This means that the content of the shared variable will be used as the address of the file to be opened.
5. Using any of the network API methods, you can change the shared variable to the address of the file you want to display. You can use a relative path or a complete URL.

The *new_ticker.svg* template may be used alternatively if you want the message to scroll through the screen when triggered.

Note that the method presented here can also be applied to permanent content, such as a score overlay on sports content by giving a large enough duration for the slide. Effects can be used in this case to render the transition visible, or removed to make a silent change.

Triggering an Alternative Content

It is possible to temporarily replace the current content by an alternative one based on the user input. The input might be based on a local (serial port or USB) or remote trigger.

To create such content, start with the procedure found in the *Remote Control of Content* on page 33. In most cases, set the normal content as the default media (the first one on the list). Each additional ID will be associated to each one of the alternative contents with their duration.

To trigger the display of an alternative content, set the control variable (default name is “menu”) to the corresponding ID. Once the media has been played, the content will revert to the normal media, which will play again from the beginning.

It might be desirable for the main media not to start over each time it is reopened. In this case, two scenarios are possible:

- The main media resume its playback, as if paused during the alternative content.
- The main media continue playing, as if playing in the background during the alternative content.

The first scenario can be controlled using the *Resume where left* parameters of the *menu_media.svg* template. Settings this parameter to 'on' will prevent the main media from restarting each time it is opened. For the second scenario, the main media must be placed in a schedule file. It is also possible to use the *spx:begin* attribute on the *<svg>* element of the main media.

Triggering of SVG Animations

For more advanced effects, you can design your custom SVG document to include animations that will be triggered remotely over the network. To do this, you design the animations as if triggered by time, but use *indefinite* as the *begin* attribute.

In a <script> element you can then instantiate a shared variable for remote control and install an update listener on the variable. From this callback, you can use the *beginElement()* or *beginElementAt()* methods of the *smil::ElementTimeControl* interface to trigger the animation.

In the sample document below, a rectangle is set to move through the screen from left to right when the “kick” variable is set to “go”.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1280 720"
dur="indefinite" viewport-fill="white">
<script><![CDATA[
var anim=document.getElementById( "anim" );
var v=createSharedVariable( "kick" );
v.addUpdateListener( function () { anim.beginElement(); } );
]]></script>
<rect x="0" y="260" width="200" height="200" fill="blue">
<animateTransform xml:id="anim" attributeName="transform"
begin="indefinite" type="translate" to="1080,0" dur="3s" fill="freeze"/>
</rect>
</svg>
```

You can also use the content of the shared variable to modify the document on the fly. In the example below, a rectangle will flash for once second each time the variable is updated and its color is set to the content of the variable.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1280 720"
dur="indefinite" viewport-fill="white">
<script><![CDATA[
var anim=document.getElementById( "anim" );
var rect=document.getElementById( "rect" );
var v=createSharedVariable( "kick" );
v.addUpdateListener( function ( sv ) {
rect.setAttribute( "fill", sv.value );
anim.beginElement();
} );
]]></script>
<rect xml:id="rect" display="none" x="540" y="260" width="200"
height="200" fill="blue">
<animate xml:id="anim" attributeName="display" begin="indefinite"
to="inherit" dur="1s"/>
</rect>
</svg>
```

Synchronized Content

Overview

A video wall is a set of screens accurately placed to create a single display. Screens may be one next to the other to compose a larger regular area, or they may have irregular positions to create a non-conventional pattern. In all of these cases, a perfect synchronization of the content displayed is required to provide a sense of uniformity, no matter how the screens are placed. The IS-SPX Player lets you control accurately when each item is displayed, enabling a very flexible solution for managing any type of video wall applications

Application Examples

- In public transportation, multiple displays are deployed along escalators. The content on the displays follows the user moving from one display to the next. This special effect increases the impact of the message to the user.
- In retails shops, hotels and bars, multiple displays are deployed in the same hall to create an ambiance or a "leitmotif": animations can be created to move from one display to the next one creating global effects.
- Two or three small size display can be grouped together to create a larger size display with higher impact to the audience.

Multiscreen Displays

IS-SPX Player's clocks are precisely time synchronized, letting the user control exactly when an item should be displayed. This feature can be used to schedule content to be displayed at a certain time of the day, or to synchronize the display of two or more IS-SPX Players. This enables the usage of the IS-SPX Player in multiscreen displays.

XPress offers an intuitive way of managing multiscreen projects, through the definition of a multiscreen project and the possibility to create virtual devices for the publication process. In this case the same project will be used on all the devices composing the multiscreen display, but each individual IS-SPX Player will display only one part of the project. This lets the user manage multiscreen displays in the same way as single-screen displays.

To display content on multiple screens using XPress, you need to perform the following operations:

- Create a project based on the multiscreen option. The multiscreen template is available from **www.amx.com**, and must be added to the *My Templates* tab under XPress.
- Create a virtual device to be able to publish your project on all the IS-SPX Players composing your multiscreen display.
- Configure each IS-SPX Player so that it knows its position in the multiscreen layout. It might also be necessary to configure the synchronization server for all of the devices.
- Ensure each device has it's multiscreen Id setup within the web.config pages.
- Publish your project on the virtual device.

Before proceeding, you will have to download and install the Multiscreen template folder to your computer if you have not already done so. For more information on installing the folder, please refer to Adding Additional Templates to Inspired Signage XPress on page 100.

Designing Basic Projects for Multiscreen Layouts

To create a basic multiscreen layout, with configurable screens, you will need to:

- Create the basic Multiscreen Project.
- Edit the index properties of your project to reflect your screen setup.
- Start working on your project.

Creating a Configurable Multiscreen Project

1. Open **My Projects** Tab in the Workspace area (FIG. 54):

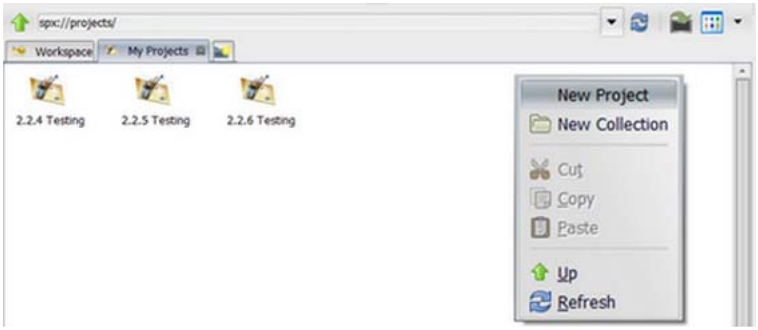


FIG. 54 New Project menu

2. Right click in the workspace area, selecting **New Project** from the popup menu.
3. A *New Project* options box appears, input <name of project>.
   a. Select the drop down option for **Format** selecting type **<Multiscreen>**
   b. Finally select the relevant **Target** type from the drop down option for the SPX product (FIG. 55):

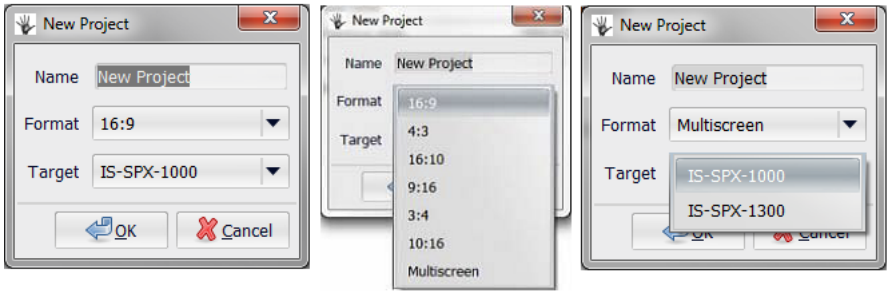


FIG. 55 New Project Options

NOTE

Compatibility of projects created for IS-SPX-1000 can be used on both target devices; however, projects created for IS-SPX-1300 will only work on this target device type, and have no backward compatibility.

4. Click **OK**, to create a basic Multiscreen Project, which will appear as named in the *My Projects* folder (FIG. 56):

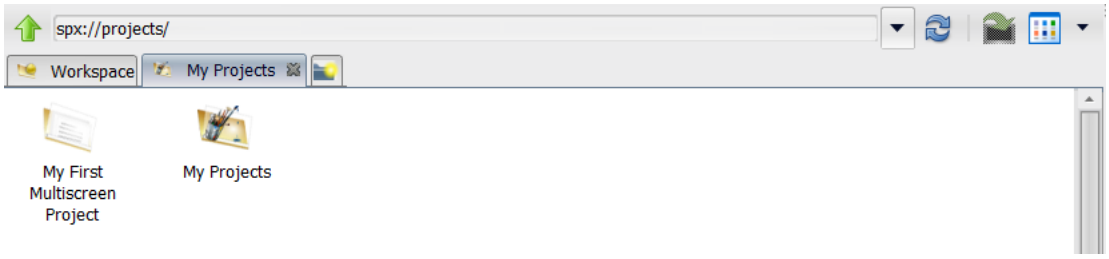


FIG. 56 Project Folder Created

5. The Basic Multiscreen Project has now been created (FIG.56), double clicking on the new project, this will open the configurable multiscreen template, showing the index.svg file in the **Workspace** area, double click the index.svg file and a preview of the 2x2 standard multiscreen and its **Properties** appear within the XPress interface in the Edit Area on the right (FIG. 57):

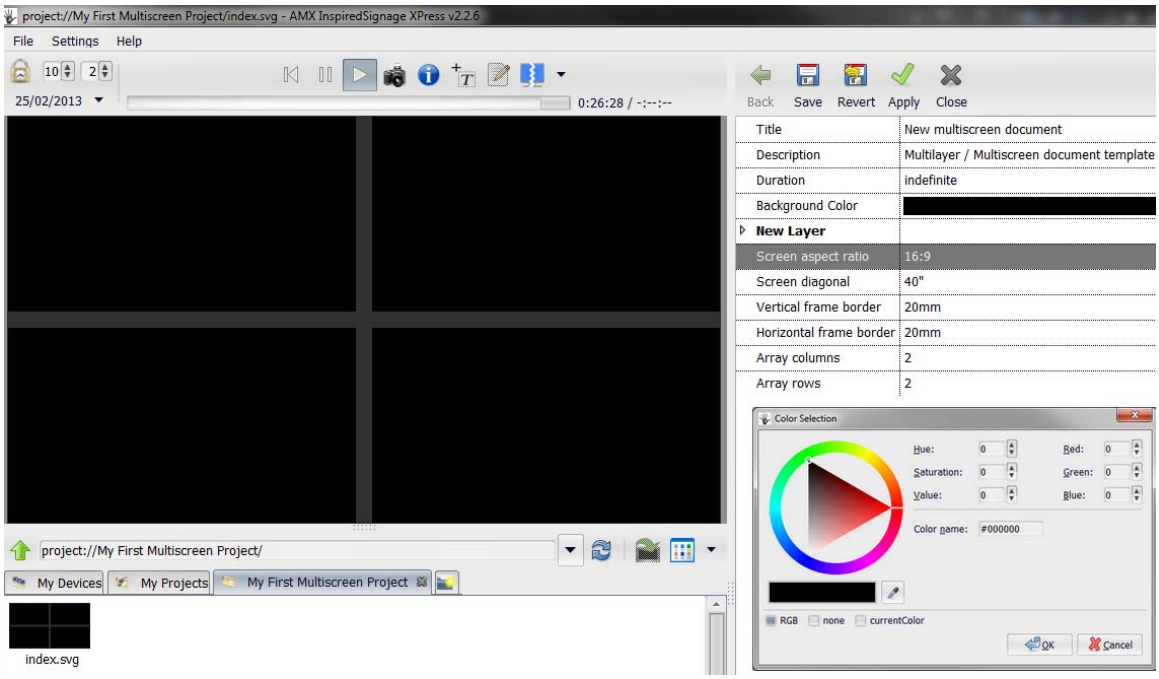


FIG. 57 Configurable Multiscreen project with Color selection options

6. Changing the information in the Fields by double clicking each input area.
   - **<Title** - enter text>
   - **<Description** - enter text>
   - **<Duration** - indefinite or media>
   - **<Background Color** - pick from the Color Selection Tool>
7. Changing the Parameters for the **New Layer** information has the same input process, with the exception of Screen aspect ratio, which is a drop down menu offering the different screen ratios (FIG. 58):

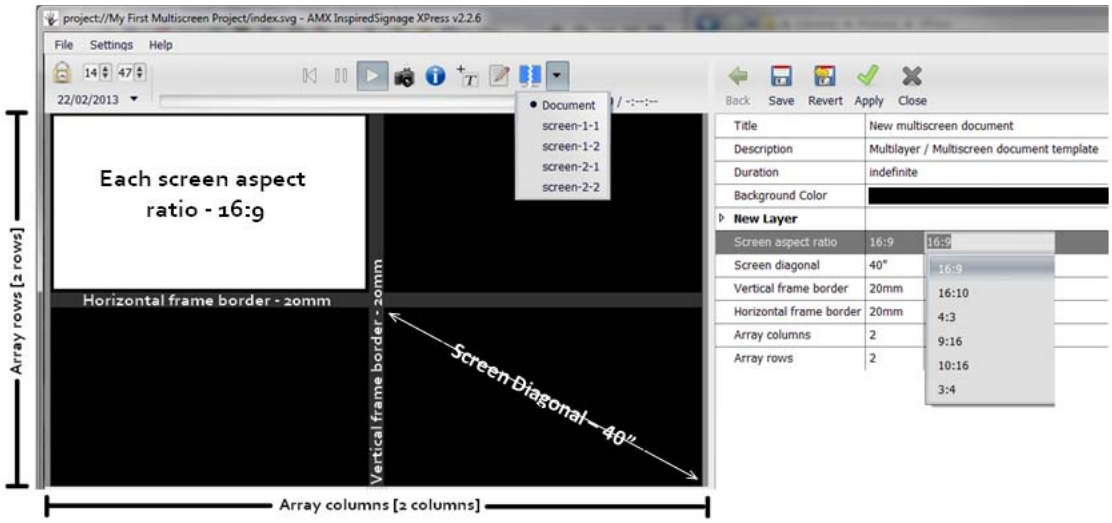


FIG. 58 Multiscreen Properties

Defaults Settings:

- **Screen aspect ratio** - 16:9, other options - 16:10, 4:3, 9:16, 10:16, 3:4
- **Screen diagonal** - 40" - reflects the dimensions of the physical screens being used
- **Vertical frame border** - 20mm - defines the frame bezel on the physical screen
- **Horizontal frame border** - 20mm - defines the frame bezel on the physical screen

Note that you can use a negative value for the frame border for overlapping projected screens.

These 3 settings are where the user needs to customize for their individual setup, and some trial and error is required to get the look and feel correct.

- **Array Columns** - 2 - refers to the number of screens the project has across
- **Array rows** - 2 - refers to the number of screens the project has down

At this point the basics of the multiscreen project have been set, and the design of content can now be added to this configurable template, by using the previously describe techniques in this guide.

NOTE

All content must obey the limitations of a single IS-SPX Player. For instance, the maximum resolution for a displayed video is SD. If you want to display video with higher resolution, the video must be cut into 4 or more segments of smaller sizes.

NOTE

When mixing IS-SPX-1000 and IS-SPX-1300 Players, the displayed content must adhere to the maximum boundaries of the IS-SPX-1000. Due to different device performance, it is recommended to use the same product range in order to keep content synchronized across multiple screens.

Creating a Multiscreen Project from Templates

The second method for working with multiscreen content is to create your XPress project and change the index.svg to one of the templates which have been downloaded from the AMX website previously.

This is done using the following steps:

1. Creating a Multiscreen for a template project will follow the same steps 1-5 from the *Creating a Configurable Multiscreen Project* section on page 46.
2. At step 5, delete the '**index.svg**' in the new multiscreen project created.
3. Open a new tab in the Workspace Area and move to the **My Templates** folder in the work area, and open the Multiscreen Folder (FIG. 59):

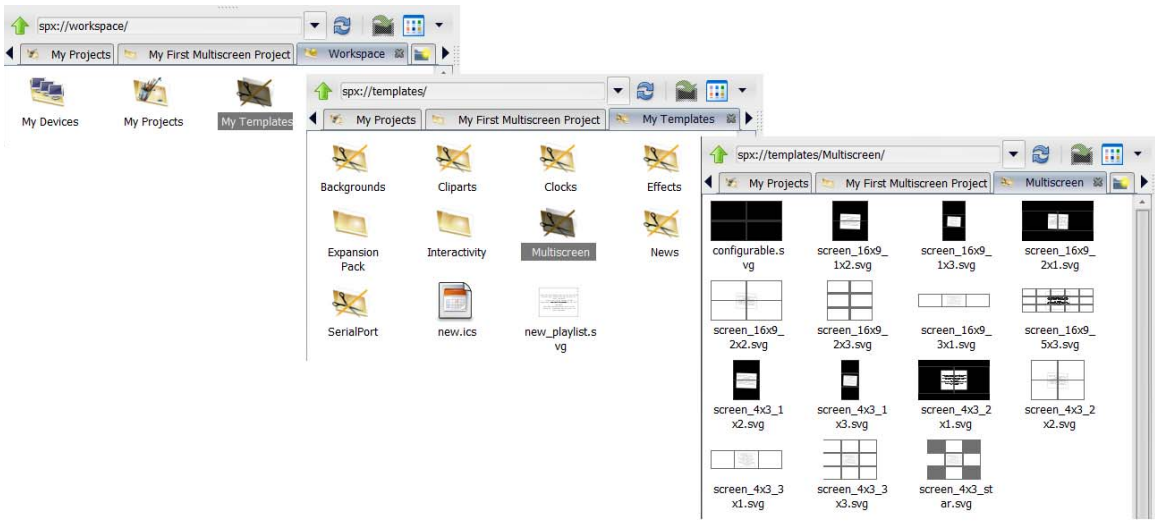


FIG. 59 Finding the multiscreen templates

4. Multiscreen templates are now visible in the Workspace Area, pick the template that the project requires, drag the template from the Multiscreen folder, hover over the project tab, which will open the new project with the blank Workspace, and drop the template into the Workspace (FIG. 60):

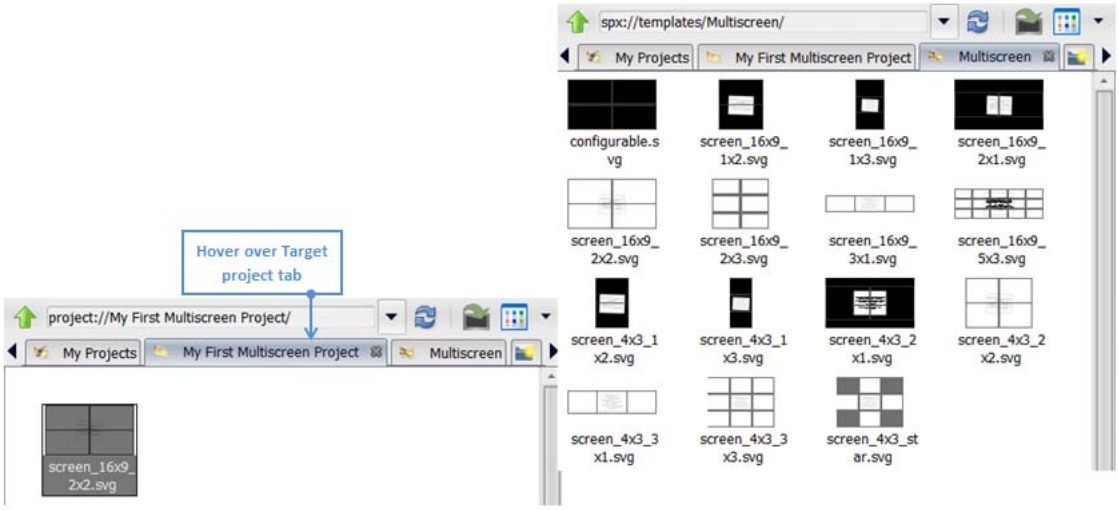


FIG. 60 Drag and drop multiscreen template to project

5. Right clicking on the new template file **screen_16x9_2x2.svg**, offers a menu to rename the file to *index.svg* which is required for the project to function correctly (FIG. 61):

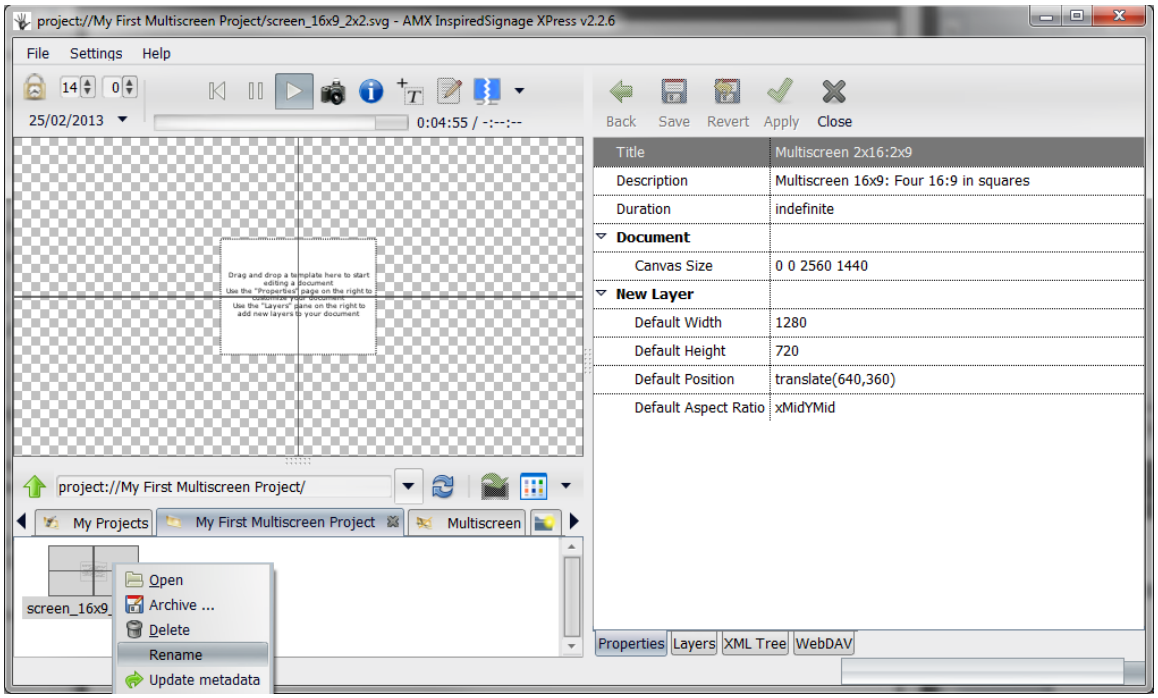


FIG. 61 Multiscreen template renaming

6. The new multiscreen project can now be **Saved** and **Edited**.

Editing Multiscreen Content

Once the index file has been created, you can start working on your project in the same way as for a single display project, by using the techniques describe in previous sections.

For instance, you can:

- Add a background covering all the displayed content.
- Add playlist over the complete displayed area.
- Add a video covering one or multiple screen.

You can choose to see a global view of your project (Document) or a single screen portion at any time by clicking the **Multiscreen layout view** button (FIG. 62):

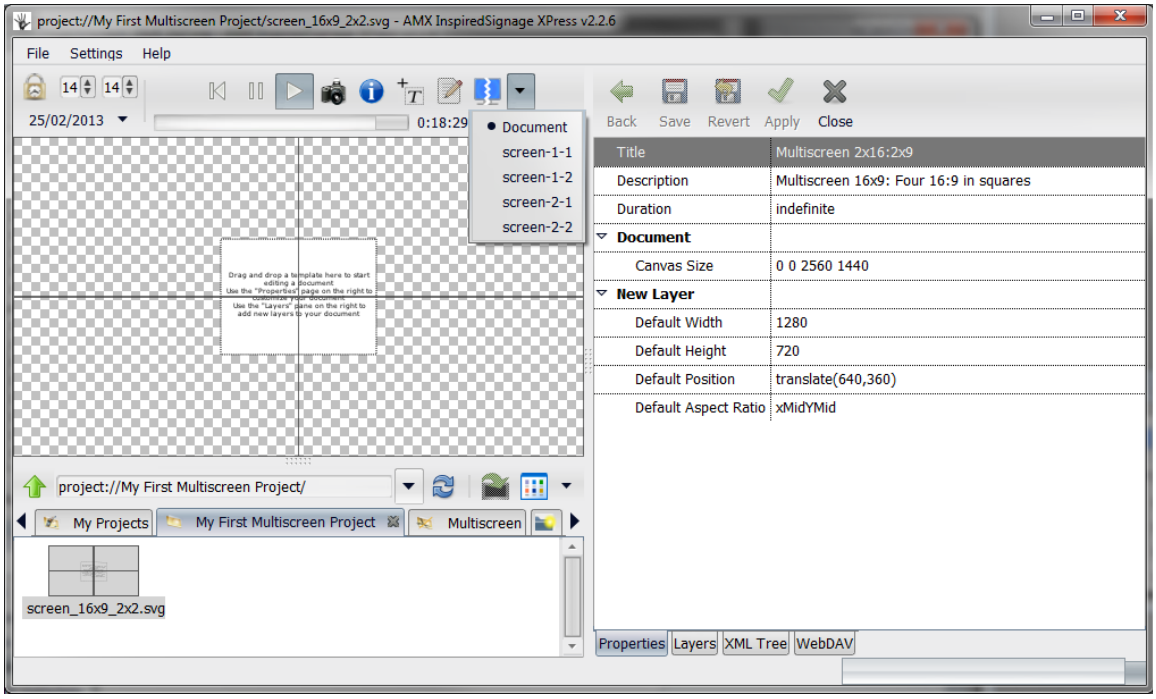


FIG. 62 Multiscreen layout drop down options

Configuring XPress and the IS-SPX Player

Before you can publish your project, you need to configure your IS-SPX Player to be ready to receive a multiscreen project. Two operations are needed: The creation of a virtual device in XPress and the configuration of the multiscreen ID of each IS-SPX Player.

Create a Virtual Device

It is possible to create group of devices under the *My Devices* folder. A project can be published to a single device or to groups of devices. This mechanism will be used to create a virtual device for multiscreen projects.

To create a virtual device:

1. Go to *My Devices* in the *Workspace* pane.
2. Create a new collection, and give it a unique name, such as "**Video Wall 9screen**" (FIG. 63):

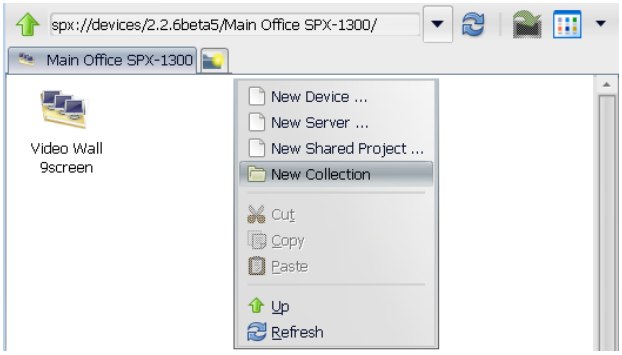


FIG. 63 Creating a New Collection of Devices

3. Find the devices to which you want to publish your project, and move them into the newly created folder. This can be achieved by drag and drop, or copy and paste (FIG. 64):

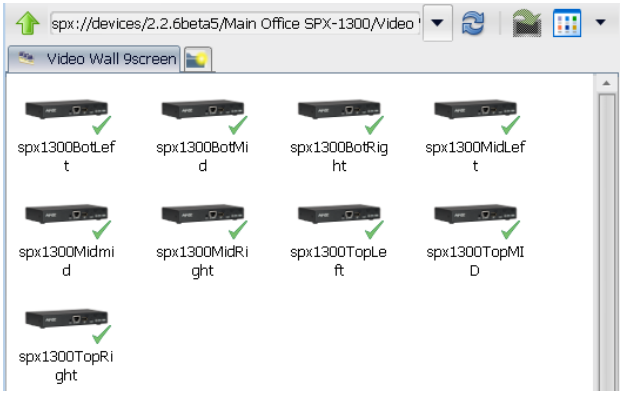


FIG. 64 Devices moved into a collection

Configure the Multiscreen ID

To configure the IS-SPX Player multiscreen ID, you need to open the HTTP configuration server of each IS-SPX Player. You can use the *Manage...* menu entry to configure each device:

1. Open the "Video Wall IS-SPX Players" folder You will need to identify each device and assign its position in the layout.
2. Right-click on the first device and choose *Manage...*. The Browser-Based Configuration Pages will open.
3. Open the *Network Settings* configuration page: The *Network Settings* configuration page for the IS-SPX-1300 is shown in FIG. 65 (the options on this page are the same for IS-SPX-1000 Players):

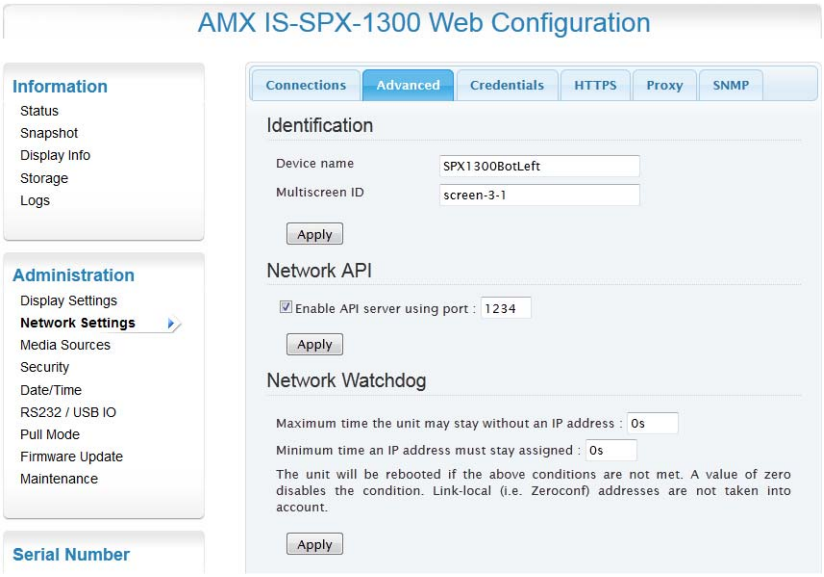


FIG. 65 IS-SPX-1300 Network Settings page - Advanced tab

4. Look for the *Multiscreen ID* under the *Identification* title. This needs to be changed to reflect the "Multiscreen Layout" screen names. e.g. screen-1-1. screen-1-2, screen-2-1, etc., depending upon screen position. For instance, inn a 2x2 multiscreen layout, the upper right is screen-1-2 and lower left is screen-2-1.

NOTE

All entries in the Multiscreen ID fields are case-sensitive.

5. This operation need to be performed for each IS-SPX Player.
6. Once all the Multiscreen IDs have been configured, you are now ready to publish your project.

Publish Your Project To Your Device

To publish your multiscreen project to your group of devices, drag and drop your project folder on top of the folder icon. Each device will be automatically updated and will display the assigned area of the video wall. Once the first project has been published, you can use the **Quick publish** button for multiscreen projects, so that changes can synchronized immediately to all players (FIG. 66).

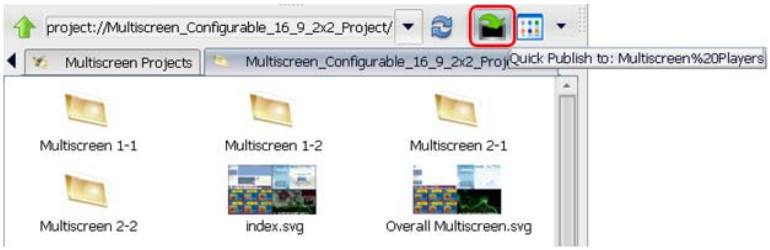


FIG. 66 Quick publishing a Multiscreen project

Synchronization

In order to be able to display content on multiple displays, the clocks of the IS-SPX Player must be synchronized. The synchronization is done using the Network Time Protocol (NTP).

Synchronizing Devices

The NTP protocol uses one or more NTP servers to be able to control the precision of the internal clock. Using the NTP protocol it is possible to reach a precision as good as 1 millisecond. This is more than sufficient to synchronize a video on multiple screens.

The IS-SPX Player comes pre-configured with some public NTP servers, which are fine to regulate local date and time, but with the following limitations:

- Time precision around 200 milliseconds.
- The firewall needs to let NTP traffic through (UDP port 123).
- Synchronized content playback across several screens requires higher precision and reliability.

When using the IS-SPX Player in a multiscreen display scenario, the time precision between the IS-SPX Player must be increased. Increasing the time precision may be done with one of these four options:

- Configure a local or ISP provided NTP server
- Use a third party hardware plug & play server
- Use a software based NTP server on a PC
- Use one IS-SPX Player as NTP server for the other IS-SPX Player.

Configure NTP Through the HTTP Interface

Open your IS-SPX Player's Browser-Based Configuration Pages and choose the *Date/Time* tab on the left, and clicking on the *Time Sources* tab. The *Data/Time* configuration page for the IS-SPX-1300 is shown in FIG. 67 (the options on this page are the same for IS-SPX-1000 Players):

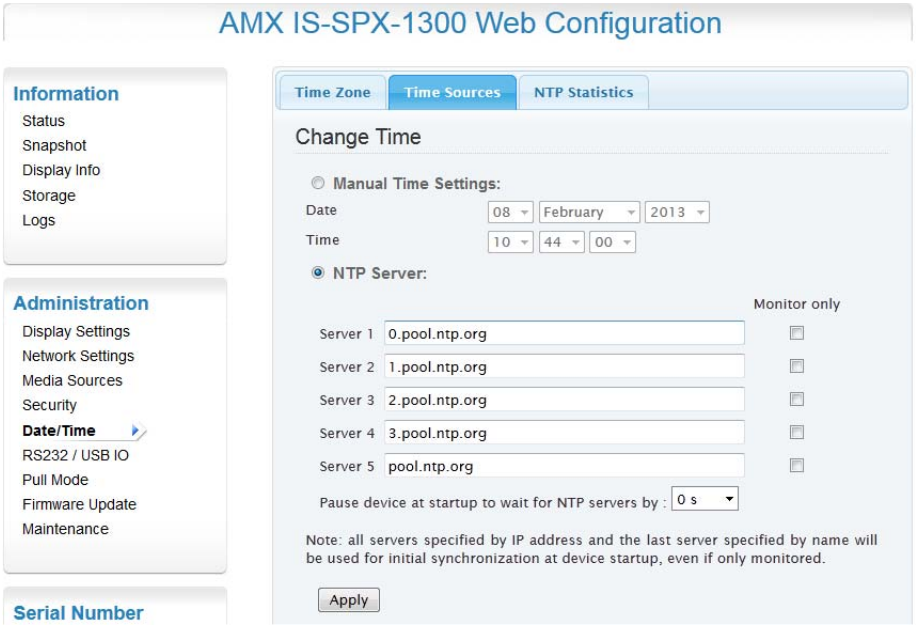


FIG. 67 Administration - Date/Time page

Once you are on the *Date/Time* configuration page, do the following:

1. Make sure the NTP Server button is selected.
2. Insert the address of your NTP server(s). Leave the *Monitor only* options unchecked. This option is needed only for advanced configurations.
3. Leave the "Pause device and startup" to "0 s". If your IS-SPX Player is using a local NTP server, you can increase this value to ensure that your device will boot after your server, in case of power loss or other circumstances.
4. Click on **Apply** to enable the new configuration

After you have configured all your devices with your local NTP server(s), they are ready to play synchronized content.

Advanced Configuration

Using an IS-SPX Player as an NTP Server

Under certain circumstances, you may want to use an IS-SPX Player as a time server for the other devices in a network:

- *You don't have a local NTP server and want to use public servers.* In this case, one IS-SPX Player will synchronize to the public servers and the other players will be synchronized to this device. This will provide better accuracy between all the devices.
- *The IS-SPX Player does not have access to the network.* In this case, one of the devices is used as master whereas the others use its time as a reference. This will not prevent your installation from time drifting, but will ensure that all the devices are synchronized.
- *You have two players that need to be synchronized.* In this case, the two devices can be connected by a cross-over cable and one IS-SPX Player configured to be the time reference for the other.

In all of these scenarios, you will have two types of devices: a master and one or more slaves. One IS-SPX Player will play the role of NTP server (master), and the others will connect to it for getting the correct time (slaves).

Before adopting this solution, please note that this is less reliable than the standard one; in particular it's important to know that the clock of the IS-SPX Players will drift apart from UTC time.

Configuration

Each device is ready to work as a NTP server, but a few parameters must be set to assure a smooth overall operation, thus execute the following procedure:

1. Designate one and only one device player as the master. It should have a static IP address. The clock should be properly calibrated before use. (For calibration information, please refer to the *Troubleshooting* on page 60.)
2. All of the other IS-SPX Players are slaves. They should also have static IP addresses, but this is not strictly required.

Configuration of the IS-SPX Players Acting As Slaves

1. Identify the “slave” IS-SPX Player and open its Browser-Based Configuration Pages.
2. In the *Date/Time* page, choose *NTP Server* and enter the IP address or URL of the “master” IS-SPX Player in the first textbox (Server 1) (FIG. 68).

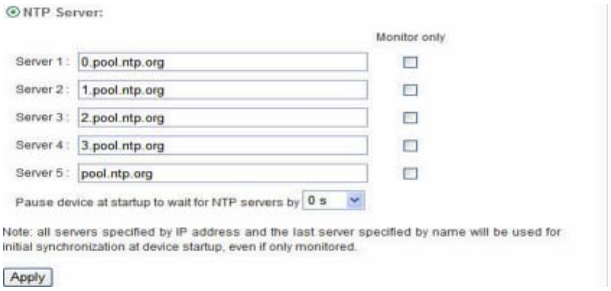

NTP Server:
Monitor only
Server 1 : 0.pool.ntp.org
Server 2 : 1.pool.ntp.org
Server 3 : 2.pool.ntp.org
Server 4 : 3.pool.ntp.org
Server 5 : pool.ntp.org
Pause device at startup to wait for NTP servers by 0 s
Note: all servers specified by IP address and the last server specified by name will be used for initial synchronization at device startup, even if only monitored.
Apply

FIG. 68 NTP Server listing

3. Select *90 s* in the menu list corresponding to the option *Pause device at startup to wait for NTP servers by.*
4. Click the **Apply** button.
5. Repeat this procedure for all your slave devices, but not the “master” (FIG. 69).

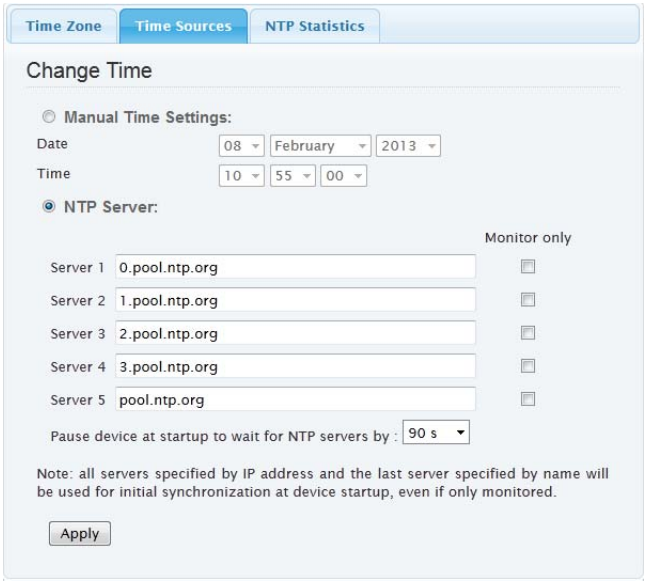


FIG. 69 NTP server configuration for slave devices

Configuration of the IS-SPX Player Acting as Master

Identify the "master" IS-SPX Player that will serve as the NTP server, and configure it as follows through the Browser-Based Configuration Pages:

1. In the *Date/Time* page, choose *NTP Server* and write the IP address of as many slave IS-SPX Players with static IP addresses as possible in the server textboxes (Server 1, Server 2, etc.) (FIG. 68).
2. Check the corresponding box(es) in the right column, under the title "Monitor only". The check box must be checked: if it is not, you will create a reference loop, and your system will not work correctly.
3. Select "0 s" in the menu list corresponding to the option *Pause device at startup to wait for NTP servers by.*
4. Click the **Apply** button.

With such a configuration, you will be able to switch on all your IS-SPX Players at the same time and have them automatically synchronized between themselves.

Master/Server Architecture Examples

As the number of IS-SPX Players in your network grows up, several master/slave configurations become possible (FIG. 70). The red arrows indicate the server address(es) entered in the master IS-SPX Player configuration (Monitor only). The black arrows refer to the values entered for the slave IS-SPX Player(s).

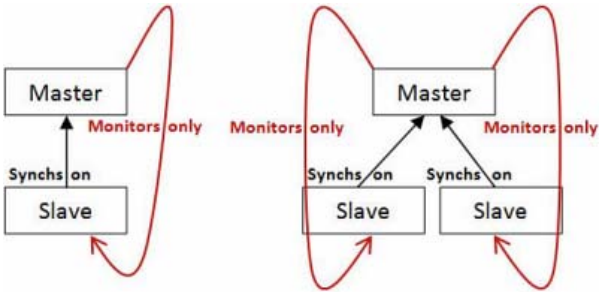


FIG. 70 NTP server/master architecture

NOTE

Multiple levels of NTP hierarchies are strongly discouraged. For large installations, it is more appropriate to invest in a NTP server.

Syntax for Multiscreen Synchronized Content

The following XML code must be implemented into the index.svg file to create a multiscreen document.

Canvas and Viewbox Concepts

A *canvas* is the complete area physically described by the SVG index file. It may have any size, but it generally corresponds to the rectangle formed by your video wall and it is usually larger than the size of one of your screens.

A *viewbox* is a subsection of the canvas, normally associated to a single screen. A viewbox can be placed anywhere inside the canvas area.

Example: in a 2x2 video wall, composed of 4 screens, the canvas will be 4 times larger than each of the 4 viewboxes associated to the 4 displays.

Multiscreen Elements

The following elements provide information regarding the canvas size and the viewboxes size and position:

```
<svg
xmlns="http://www.w3.org/2000/svg"
xmlns:spx="http://www.spinetix.com/namespace/1.0/spx"
width="100%" height="100%"
viewBox="xc yc wc hc" >
<spx:multiScreen>
<spx:screen viewBox="x y w h" xml:id="screenID"/>
</spx:multiScreen>
[...]
</svg>
```

Value definitions:

- *viewBox="xc yc wc hc"* is the space position and full size of the canvas, in pixels, where:
  xc is the x-axis coordinate of the top-left point of the canvas,
  yc is the y-axis coordinate of the top-left point of the canvas,
  wc is the total width of the canvas,
  hc is the total height of the canvas.
- *<spx:multiScreen> </spx:multiScreen>* is the element enabling the multiscreen functionality.
- *spx:screen viewBox="x y w h" xml:id="screenID"/* is the element representing each of the players that compose the complete canvas. It's attributes are:
  x is the x-axis coordinate of the top-left point of this screen,
  y is the y-axis coordinate of the top-left point of this screen,
  w is the width of this screen,
  h is the height of this screen,
  screenID is the screen identification name of your choice.

Synchronization Elements

Two possible techniques are available to ensure a correct synchronization between content on different screens. These include the use of a schedule file, or the usage of an in-line beginning time of a document. The predefined multiscreen templates use the *inline begin time* strategy. This means that synchronization will be automatic when those templates are used.

The iCalendar (.ics) File

The first option for synchronizing content displayed on multiple screens is using an iCalendar file to contain the multiscreen file. The structure of your project should be as follows:

- Create an *index.svg* with your multiscreen setup. The file should contain the *<spx:multiScreen>* instructions.
- Create a iCalendar schedule file.
- Add the content designed for multiscreen use into your schedule.

This configuration let you use multiple schedules together with multiscreen content. You can use your custom SVG file or one of the multiscreen templates as your index file.

If you want to use your multiscreen in a setup combining synchronized and unsynchronized content, the following alternative approach is also possible:

- Create an *index.svg* file using the aspect ratio of a single screen.
- Create a iCalendar schedule.
- Create multiscreen files with the <*spx:multiScreen*> instructions and add them to the calendar.
- Create single screen content and add them to the calendar. Note that you can use different calendars depending on the IS-SPX Player (using the switch of JavaScript, for instance). In this latter case, the content to be synchronized between multiple screens must start and end at the same time in the different calendars.

The spx:begin Element

Instead of using a complete calendar file just for the synchronization of the multiple screens, it is possible to use an in-line reduced schedule inside the index file. The schedule is described using a proprietary tag in the svg element: *spx:begin*. The *spx:begin* describes the starting point of the document. Using this attribute has the same effect as putting the document in a schedule file starting at the same time as the one described by the attribute.

The spx:begin element can be used as follows:

```
<svg
xmlns="http://www.w3.org/2000/svg"
xmlns:spx="http://www.spinetix.com/namespace/1.0/spx"
height="100%" width="100%"
spx:begin="Thhmmss"
viewBox="xc yc wc hc" >
[...]
</svg>
```

Value definitions:

- *spx:begin="Thhmmss*" is the presentation beginning time, where:
  T is a fixed flag,
  hh is the hour of the day,
  mm are the minutes,
  ss are the seconds.

This feature is used by all the predefined multiscreen templates.

Additional Syntax for Multiscreen Content

The following two methods are also available to build multiscreen contents:

- Extension of the SVG <switch> element
- JavaScript MULTI_SCREEN_ID variable

Extension of the SVG <switch> Element

The SVG <switch> element allows you to conditionally display SVG elements. This is a standard SVG element described in the SVG specification. For the handing of multiscreen content, a proprietary condition has been added. This lets you visualize a specific content based on the screenID of the player running the project. The <g> content is shown only if *spx:multiScreenId* is equal to “screenID”. Multiple <g> contents, for different “screenID” values can be listed into the same <switch> element.

```
<switch>
<g spx:multiScreenId="screenID">
</g>
<switch>
```

Value definitions:

- *spx:multiScreenId="screenID"* is the variable tested by the switch, it contains the ID of the current screen.
  screenID is the value associated with each of the screens where the content will be displayed (e.g. “left”, “right”, “top”, “bottom”, etc).

Extension of JavaScript

When using JavaScript in an SVG file, the variable MULTI_SCREEN_ID is always defined. It allows you to visualize a specific content based on the screenID of the player running the project.

Value definitions:

- *MULTI_SCREEN_ID="screenID"* is the variable tested by the if statement, it contains the ID of the current screen.

  screenID is the value associated to each of the screens where the content will be displayed (e.g. “left”, “right”, “top”, “bottom”, etc).

Example: Screen Layout With Six Screens

The following example creates a 2-columns, 3-rows grid with 6 16:9 screens set horizontally

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:spx="http://www.amx.com/namespace/1.0/spx"
height="100%" width="100%"
spx:begin="T000000" viewBox="0 0 2560 2160" >
<spx:multiScreen>
<spx:screen viewBox="0 0 1280 720" xml:id="TopLeft"/>
<spx:screen viewBox="1280 0 1280 720" xml:id="TopRight"/>
<spx:screen viewBox="0 720 1280 720" xml:id="MidLeft"/>
<spx:screen viewBox="1280 720 1280 720" xml:id="MidRight"/>
<spx:screen viewBox="0 1440 1280 720" xml:id="BotLeft"/>
<spx:screen viewBox="1280 1440 1280 720" xml:id="BotRight"/>
</spx:multiScreen>
```

Example: Screen Layout With an Empty Border Between the Screens

To obtain a better total effect, you should take into account the space that may remain unusable between the displays, due to the screen frames. In this example, we consider a thickness of 80 pixels for the screen frames. To create the final layout, you must sum the internal borders thickness to the viewbox parameters, both in the (*x*,*y*) position and in the (w)idth, (h)eight sizes.

The following example creates the same grid as the previous one, but empty internal borders are also calculated:

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:spx="http://www.amx.com/namespace/1.0/spx"
height="100%" width="100%"
spx:begin="T000000" viewBox="0 0 2640 2320" >
<spx:multiScreen>
<spx:screen viewBox="0 0 1280 720" xml:id="TopLeft"/>
<spx:screen viewBox="1360 0 1280 720" xml:id="TopRight"/>
<spx:screen viewBox="0 800 1280 720" xml:id="MidLeft"/>
<spx:screen viewBox="1360 800 1280 720" xml:id="MidRight"/>
<spx:screen viewBox="0 1600 1280 720" xml:id="BotLeft"/>
<spx:screen viewBox="1360 1600 1280 720" xml:id="BotRight"/>
</spx:multiScreen>
```

Example: Screen Layout With “Spread Displays”

A completely free pattern may be adopted for your screens: empty spaces between them are possible and custom positions can be assigned. Screen contents can also overlap, increasing the sense of continuity between displays. In this example, we consider a wide area with some screens only covering a partial area of the whole space and with an overlapping corner.

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:spx="http://www.amx.com/namespace/1.0/spx"
xmlns:xlink="http://www.w3.org/1999/xlink"
height="100%" width="100%"
spx:begin="T000000" viewBox="0 0 4260 1920" >
<spx:multiScreen>
<spx:screen viewBox="50 50 1280 720" xml:id="TopLeft"/>
<spx:screen viewBox="1706 240 1280 720" xml:id="TopRight"/>
<spx:screen viewBox="852 1150 1280 720" xml:id="BotLeft"/>
<spx:screen viewBox="2800 800 1280 720" xml:id="BotRight"/>
</spx:multiScreen>
```

Example: <switch> SVG Element

The following example shows the use of the standard SVG <switch> element to display different contents on a screen based on its screenID (e.g. “left, “right”, “top”, “bottom,” etc.).

```
<spx:multiScreen>
<spx:screen viewBox="0 0 1280 720" xml:id="left"/>
<spx:screen viewBox="0 0 1280 720" xml:id="right"/>
</spx:multiScreen>
<g display-align="center" font-family="Arial" font-size="100"
text-align="center" transform="translate(320,180)">
<switch>
<g spx:multiScreenId="left">
<textArea fill="#fed600" height="360" width="640" >I'm left screen!
</textArea>
</g>
<g spx:multiScreenId="right">
<textArea fill="#00d6fe" height="360" width="640" >I'm right screen!
</textArea>
</g>
<g>
<textArea fill="#d600fe" height="360" width="640" >No screen defined
</textArea>
</g>
</switch>
</g>
```

Example: JavaScript

The following example shows the usage of the JavaScript MULTI_SCREEN_ID variable to control the text content of a textArea item.

```
<spx:multiScreen>
<spx:screen viewBox="0 0 1280 720" xml:id="left"/>
<spx:screen viewBox="0 0 1280 720" xml:id="right"/>
</spx:multiScreen>
<script>
<![CDATA[
if (MULTI_SCREEN_ID == "left")
document.getElementById("textId").textContent="I'm left!";
else if (MULTI_SCREEN_ID == "right")
document.getElementById("textId").textContent="I'm right!";
else
document.getElementById("textId").textContent="No screen defined!";
]]>
</script>
<g>
<text x="50" y="50" xml:id="textID">_Text_will_be_here_</text>
</g>
```

Troubleshooting

Synchronized content over multiple screen needs a correct configuration of the IS-SPX Player and a well-designed project to work as expected. In the event that the project is not displaying as desired, the following is a list of parameters to check before contacting AMX Technical Support.

NTP Synchronization

The content are not synchronized, i.e. one or more screens play their part of media earlier or later than the other displays. Please check the following:

- Make sure that NTP is enabled through the Browser-Based Configuration Pages, and that you have chosen one or more NTP servers:

  It should be on your local network in order to be able to achieve the few milliseconds of precision required.

  Public NTP servers are usually too “far” to provide such a precision. Please refer to the *Synchronization* on page 52 for details on inserting NTP server addresses.
- Verify your IS-SPX Players NTP statistics:

  Values in the column “Time offset” should be as close to 0 as possible.

  Make sure that at least one NTP server is always reachable (at least one column “Reach” should be equal to 100%).

In case your NTP server can't be reached (“Status” column remains always “Initializing”), it may be behind a firewall or just wrongly configured. Check the NTP server configuration with your system administrator. For further details on the NTP protocol, please refer to **http://www.ntp.org**.

If you used the master/slave configuration, be careful to correctly implement the architectures covered in the *Using an IS-SPX Player as an NTP Server* on page 54. Double check your master ("Monitor only" selected) and slave configurations ("Monitor only" NOT selected).

Project Creation

Verify that your index.svg file contains the *spx:begin* attribute, or that your project uses a schedule file (.ics), as explained in the *Syntax for Multiscreen Synchronized Content* on page 56. A good test to verify if your content uses a begin attribute or a calendar is to open the *index.svg* file in XPress. If your project starts from the beginning, then no schedule is present.

If opening the project multiple times always shows the same starting point, then no schedule is present and the playback will not be synchronized.

Internal Clock Calibration

The local clock on the device is not calibrated when shipped from the factory. NTP behaves differently when the local clock has not yet been calibrated, and it will not synchronize as expected. Figure 20 below shows the configuration window before calibration is completed.

Calibration consists in finding the offset in frequency between the internal clock and a NTP reference and is necessary for normal operation of NTP.

The calibration process is started automatically and will take approximately 60 minutes when at least one NTP server (other than *loopback*) is available. Once the calibration data is saved, it is reused when the device is rebooted or powered off.

To calibrate the clock of an IS-SPX Player:

1. Start the player.
2. In your Web browser, open the player's Browser-Based Configuration Pages.
3. Look at the *NTP statistics* section of the *Date/Time* page. If the end of the *NTP statistics* section states that the clock is calibrated, no more operations are required. If the clock is not calibrated, continue this procedure.
4. Check that at least one NTP server other than loopback is reachable and selected as peer.
5. Allow for some time in order for reachable servers to become peers.
6. If no NTP servers are reachable, check that the NTP servers are correct and that the network configuration is working. If not, modify the settings and restart the procedure.
7. Once a peer other than *loopback* is selected, leave the device running for at least 70 minutes without rebooting.
8. After 70 minutes, the local clock should be calibrated. Reboot the device from the Browser-Based Configuration Pages. Do **NOT** unplug the device before rebooting.

Interactivity

Overview

In addition to static control of displays through an IS-SPX Player, Inspired Signage XPress allows interactivity between the device and an end user. With the use of AMX touch screens and other touch-enabled displays, you can create interactive banners, menus, and other presentations that encourage user involvement.

Creating Interactive Content

Creating interactive content through XPress, such as creating a menu or controlling displayed media, uses a combination of JavaScript programming and the use of templates designed for this purpose. These templates are available at **www.amx.com**. Creating interactive content for the IS-SPX Player requires configuring the device to allow enabling events and configuration of touch screens.

NOTE

Before proceeding, you will have to download and install the Interactivity template folder to your computer if you have not already done so. For more information on installing the folder, please refer to Adding Additional Templates to Inspired Signage XPress on page 100.

Interactivity Template Files

Open the *Interactivity* folder in the *My Templates* tab of the Workspace pane to view the individual files. These files include:

- keyboard.svg - This file presents a configurable touchscreen keyboard.
- keystroke.svg - This file controls the reactions between the keys in a keyboard and the display.
- **menu_buttons.svg** - This file allows a response when a button on a touch screen is pressed.
- **menu_media.svg** - This file allows control of the presented media in a menu listing.
- **menu_select.svg** - This file allows selection between the different entries in a menu.
- **playback.svg** - This file allows control of variables to control pauses and playbacks.
- **queuing.svg** - This file allows control of a controllable counter, such as a "Now Serving" sign.
- **slider.svg** - This file allows control of video files and animations through a draggable slider bar.

The properties of each file may be displayed by clicking on the file and viewing the options in the *Edit Mode* Workspace. The tabs at the bottom of the Workspace display information on each of the components of the particular file (FIG. 71).

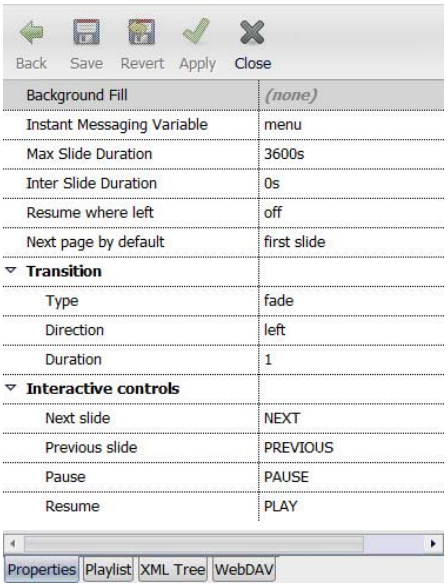


FIG. 71 Edit Mode Workspace

The names of the files in the Interactivity folder may be changed and the file will still function normally in XPress. However, make sure either to use intuitive names for each file, or document the name changes and keep the information in a safe place. If the names of SVG files are changed, it can be difficult to ascertain their function later.

Creating a Touch Screen Menu

Now, let's begin with creating a basic touch screen menu. The menu will not have too many features: upon pushing a button on the touch screen, the final project will start the display of media. That media can be an item on part of the screen, a video on the full screen, or another menu. You will create the buttons using the *menu_buttons.svg* file, select the media using *menu_media.svg*, and test the result in XPress, using the Interactivity mode.

Creating Buttons

To start, create a new folder in the **My Projects** tab, called *Touch Screen*. The only file in this folder will be *index.svg*.

1. Double-click on *index.svg* to open it in the Workspace.
2. Click **Save**, and choose a location for the files to reside.
3. Open the *Interactivity* folder in *My Templates* and drag the file *menu_buttons.svg* onto the Workspace. A button will appear in the Workspace, and a *menu_buttons.svg* entry appears in the *Buttons* tab of the *Edit Mode* Workspace (FIG. 72).

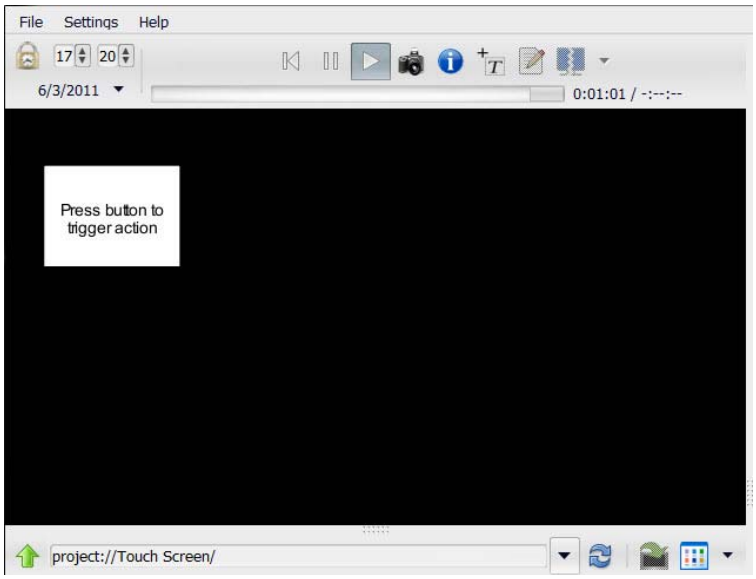


FIG. 72 Buttons appearing in Workspace

4. Buttons can be made from images, text, SVG files, or even videos. From the folder in *My Templates*, open the *Cliparts* folder and then the *Arrows* folder. Drag and drop the *E Red.svg* file over the button in your project (FIG. 73).

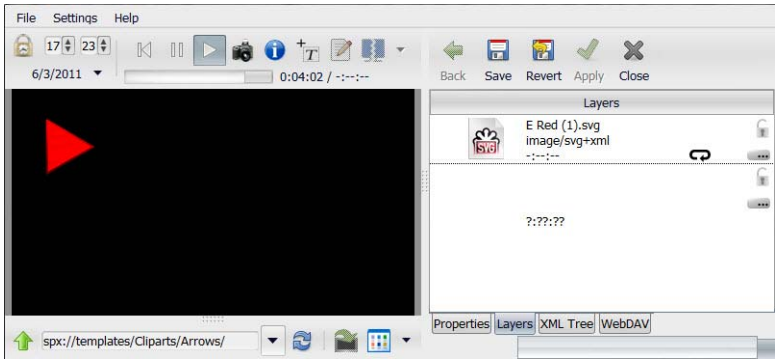


FIG. 73 Arrow on button

5. Click on the *menu_button.svg* file and add an action ID to the button (FIG. 74). The default in this case is *P1*, but the Action ID may be any name, so long as it is unique to this project. In this case, enter *One*.

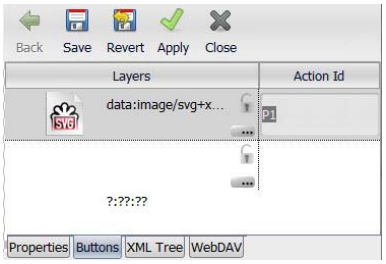


FIG. 74 Adding an Action ID

6. Save your changes.

Controlling Media

Now that the basic media files have been added, we need to add a way to control them. To start, from the *Interactivity* folder, drag and drop the *menu_media.svg* file into your project.

1. For this project, open the *Cliparts* folder in the *My Templates* tab of the Workspace, and select *E Green.svg*.
2. Add your media files to the project. Again, these can be image, video or SVG files. The first media file is the default one, and once it has been played, the document returns to the default file. In this case, drag and drop the *E Green.svg* file from the *Arrows* folder to your project (FIG. 75).

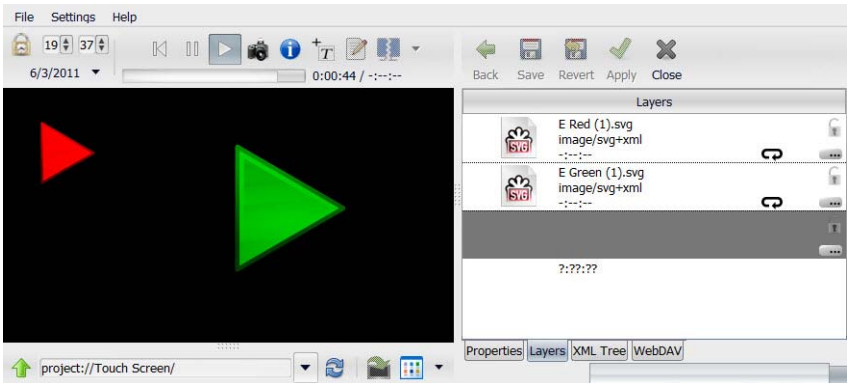


FIG. 75 Arrow on menu_media file

3. Click on the *menu_media.svg* file and then click on the *Playlist* tab. Click on the *Page Id* pane to enter a Page ID for the *E Green.svg* file (FIG. 76). As with the Action ID put in previously, enter *One*. (Note that the Page ID must precisely match the Action ID with which it is associated.)

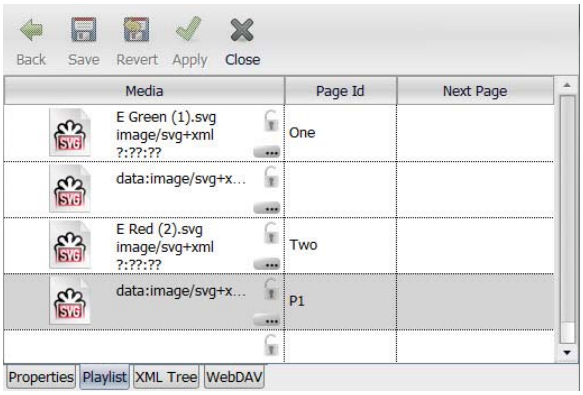


FIG. 76 Page ID

4. Click on the **Save** button to save your changes.

Testing the Project

Now, enable the Interactivity Mode in XPress. This is done by clicking the **Enable Interactive Mode** button in the *Animation Controls* navigation bar (FIG. 77).

FIG. 77 Animation Controls navigation bar

By enabling Interactive Mode, you cannot edit anything within the project until you leave it. At this point, mouse and keyboard actions are sent to the project. Click the button again to enable editing (FIG. 78).

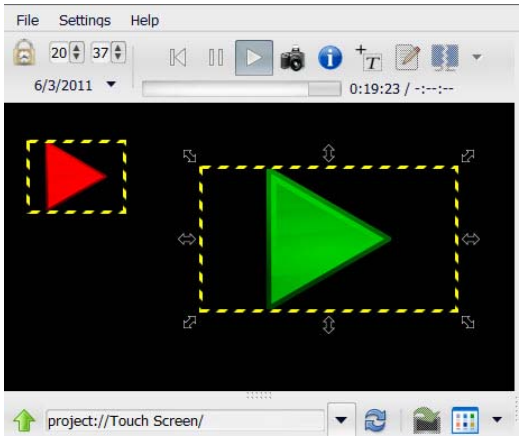


FIG. 78 Editing enabled

To test the content of your project:

1. Enter *Interactive Mode*.
2. Click on the button in the project.
3. In the *Properties* tab, set the *Instant Messaging Variable* to the previously configured Action ID (FIG. 79).

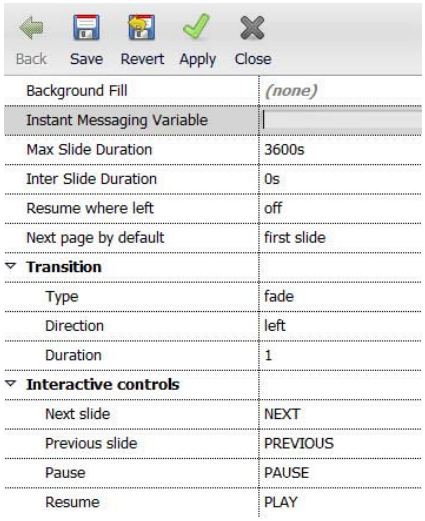


FIG. 79 Instant Messaging Variable

4. The media file with the corresponding Action ID is now enabled. In Interactive Mode, clicking on the small button should change the color of the larger arrow.

Creating a Keyboard Menu

Keyboard menus offer an additional level of interactivity, as these allow particular keys on a keyboard or remote control to select a particular menu entry. As with other menus, selecting a menu entry triggers a media file. Creating a keyboard menu is a four-step process that requires:

- Creating the menu, using the *menu_select.svg* template file.
- Mapping keystrokes to menu actions, using the *keystroke.svg* file.
- Selecting the media, using *menu_select.svg* again.
- Testing the results in XPress, using the Interactivity Mode.

Creating the Menu

To create the menu:

1. Create a new project in the *My Projects* tab. For this exercise, name the new project "Keyboard Menu".
2. From the *Interactivity* folder in the *My Templates* tab, drag the *menu_select.svg* file into your project (FIG. 80). The document size will adapt to the area.

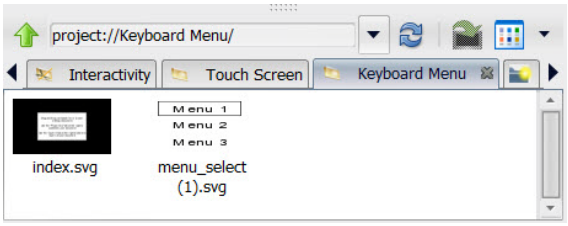


FIG. 80 Keyboard Menu - menu_select.svg

3. Double-click the *menu_select.svg* file to open it (FIG. 81).

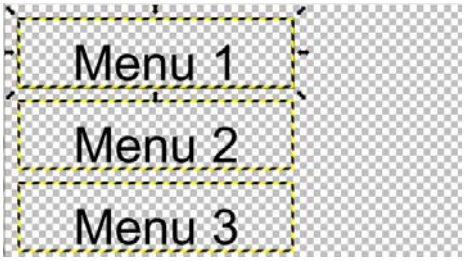


FIG. 81 Details of menu_select.svg

4. The file is now ready to add menu items to the project. These may be images, text, SVG files, or even videos. For this exercise, go to the *My Templates* tab in the Workspace, open *Cliparts*, open *Arrows*, and add three different arrows to the file (FIG. 82).

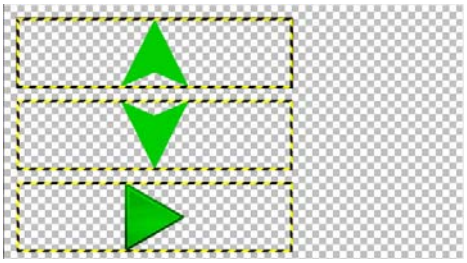


FIG. 82 Adding arrows to menu_select.svg

5. In the *Edit Mode* Workspace, click the *Menu* tab. Note that along with the *Action ID* field, each button now has fields for *Left*, *Right*, *Up*, and *Down* (FIG. 83).

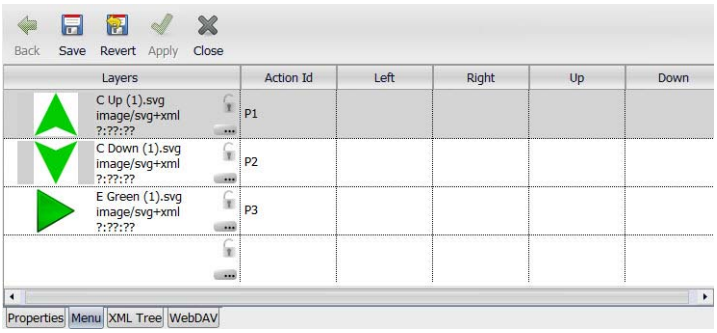


FIG. 83 Button Action IDs

6. Add an Action ID to each button. This may be any letter or number combination, but should be unique to the button.
7. Configure the display order. By default, the *next* and *previous* actions are defined, but the actions for *Left*, *Right*, *Up*, and *Down* may be configured at this time as well.

Configuring the menu

Now is the time to configure the menu for specific effects. To do so:

1. Select the *menu_select.svg* file.
2. In the *Edit Mode* Workspace, click the *Properties* tab (FIG. 84).

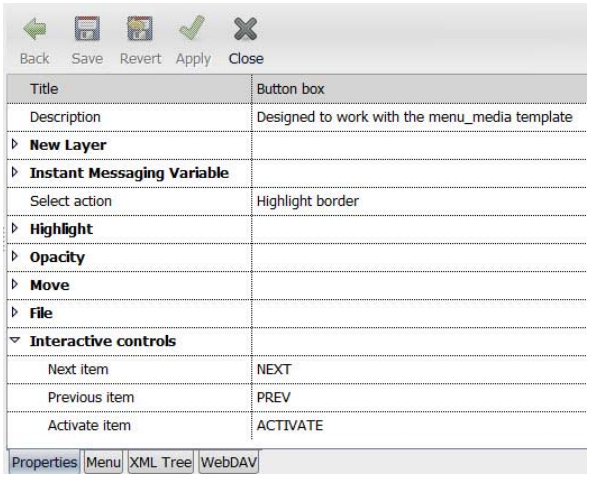


FIG. 84 Properties tab

3. Click the arrows next to each of the main categories to expand them (FIG. 85).

Title	Button box
Description	Designed to work with the menu_media template
New Layer	
Default Width	320
Default Height	180
Default Position	translate(16,16)
Default Aspect Ratio	xMidYMid
Instant Messaging Variable	
Menu selected	menuSelect
Menu activated	menu
Change menu	menuChange
Select action	Highlight border
Highlight	
Color	
Opacity	1
Width	3
Opacity	
Default	0.5
Selected	1
Move	
Horizontal offset	0
Vertical offset	0
Zoom factor	1
File	
Postfix	_select
Interactive controls	
Next item	NEXT
Previous item	PREV
Activate item	ACTIVATE

Properties | Menu | XML Tree | WebDAV

FIG. 85 Properties tab - expanded

In this instance, let's consider changing the features of the menu itself. Each button in the menu is currently surrounded with a black border and a clear background, which doesn't stand out in front of the black main background. Under the *Properties* tab:

1. Under *Highlight*, double-click the field next to *Color* to open the *Color Selection* pane (FIG. 86).

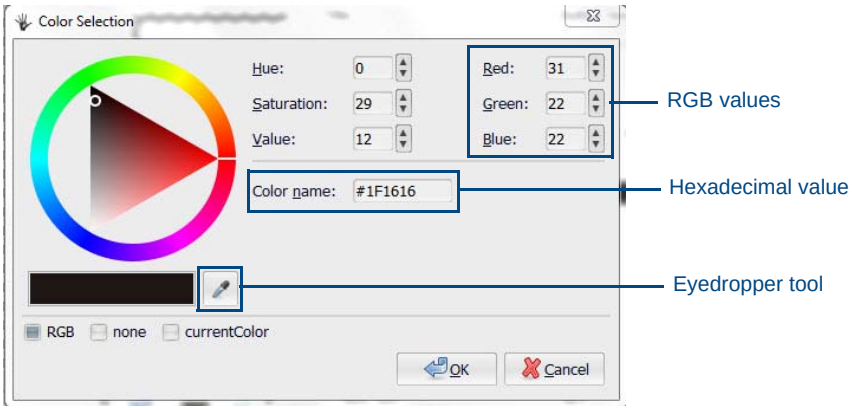


FIG. 86 Color selection pane

2. Pick a pleasing but contrasting color. This may be done by clicking on the color wheel, or using the Eyedropper tool to select a comparable color from another element in the project. If you know the exact color you want, you can enter the RGB values in the *Red*, *Green*, and *Blue* fields, or enter the hexadecimal value in the *Color name* field. For this project, enter *#00BC00* into the *Color Name* field and click **OK** (FIG. 87).

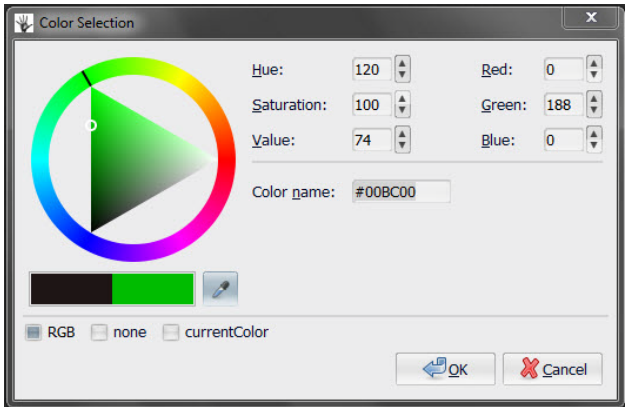


FIG. 87 Color Selection - green

NOTE

The selected color is also the color used in the E Green.svg arrow button. This can be selected by using the Eyedropper tool to touch the button and add its color to the Color Selection wheel.

3. Also under *Highlight*, note the *Opacity* and *Width* fields. The *Opacity* field controls the opacity of the border compared to its background: a higher number means a more opaque border. The *Width* field controls the total width of the border: the higher the number, the wider the border around the button. In this example, leave the *Opacity* alone, but lower the *Width* to 2.
4. Next, under *Move*, note the *Horizontal offset*, *Vertical offset*, and *Zoom factor* fields. These can control the displacement of the buttons by one pixel when clicked. In this example, the currently selected *Zoom factor* will expand the button by one pixel along its entire range.
5. When finished, click the **Save** button at the top of the *Edit Mode* Workspace to save your changes.

Controlling the Menu

Controlling the menu requires the use of *shared variables*. These shared variables may be used to exchange information between SVG files, and may also be used to exchange information through an entire network.

NOTE

For more information on shared variables, please refer to Shared Variables on page 85.

For this exercise, three variables are used for menu control:

- **Menu selected:** This variable affects the file changed when an item is selected.
- **Menu activated:** This variable affects the file changed when an item is activated.
- **Change menu:** This variable controls the state of the menu.

These variables may be found in the *Edit Mode* Workspace, under *Instant Messaging Variable* (FIG. 88).

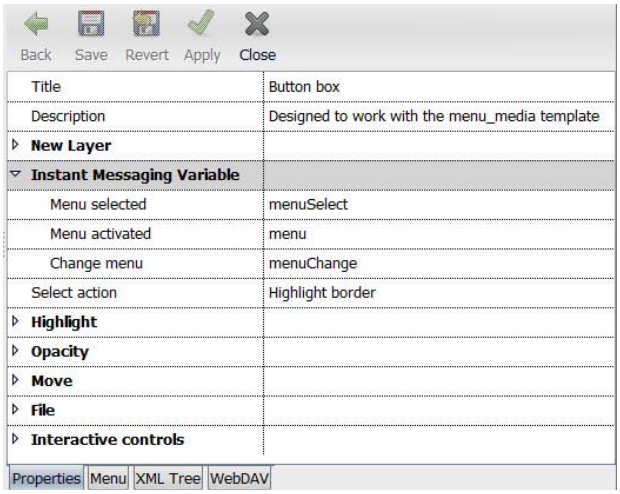


FIG. 88 Instant Messaging Variable

Next, open the *Interactive controls* section (FIG. 89). This section contains three additional variables: *Next Item*, *Previous Item*, and *Activate Item*.

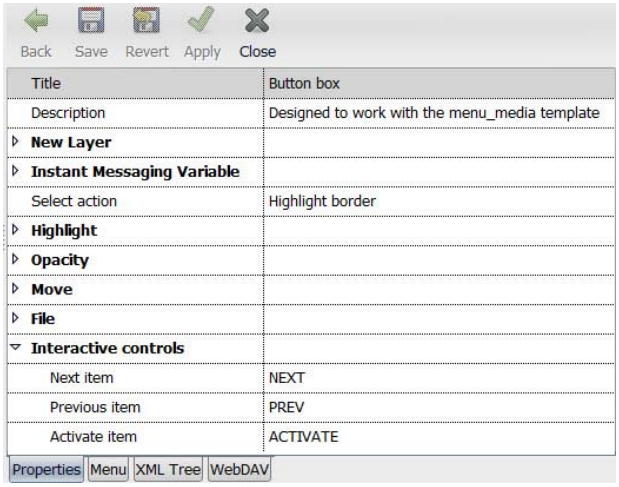


FIG. 89 Interactive controls

These three controls may be used to control displayed media files in the menu. In place of NEXT, PREV, and ACTIVATE, enter the Action IDs of the files to be controlled.

Keystrokes

By default, menus are controlled by a shared variable. In order to control a menu through a keyboard, you need to be able to map a key to a particular action. For instance, mapping the arrow keys on a keyboard might allow users to move through entries on a menu, or particular menu shortcuts might be mapped to a specific character key.

To map keys to an action:

1. From the *Interactivity* folder of the *My Templates* tab, drag the file *keystroke.svg* into the Keyboard Menu project Workspace.
2. Open the *index.svg* file and click the *Layers* tab in the *Edit Mode* Workspace if it is not already selected.
3. Drag and drop the *keystroke.svg* file into the *Layers* tab (FIG. 90).

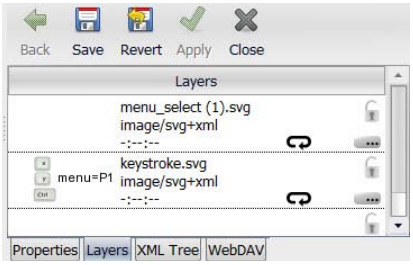


FIG. 90 Layers tab - keystroke.svg

4. Click on the *keystroke.svg* layer to open it and click on the *Key Identifiers* tab (FIG. 91).

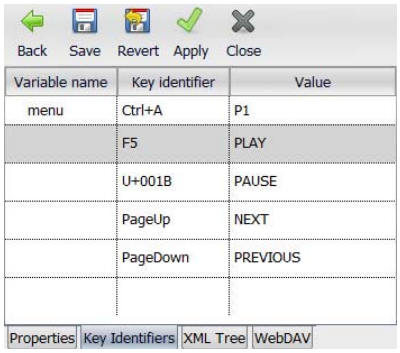


FIG. 91 Key Identifiers tab

5. In the *Key Identifiers* tab, you will notice that only one variable name is listed, but multiple key identifiers and values. Each key needs to have a value assigned to it. Variable names, key identifiers, and values may be changed by double-clicking on the entry and adding the new value in the field.
6. To add a new row for variables, key identifiers, or values, right-click on any cell and select *Insert*. The new row may be dragged to a new location, or may be moved by right-clicking on it and selecting *Move Up* or *Move Down*.

NOTE

More than one variable may be added to the Key Identifiers *tab. Add each new variable below the previous one by inserting new rows.*

Controlling Media

Now that the keystrokes are mapped, the next step is to connect them to actual functions. To do so:

1. From the *Interactivity* folder in the *My Templates* tab, drag and drop the *menu_media.svg* file into the *Keyboard Menu* project.
2. Add your media files to the project, as layers in the *index.svg* file. These may be images, video files, or SVG files. The first one added is the default media file: once another media file has finished playing, the project returns to the default.
3. Double-click the *menu_media.svg* file, and enter the Page IDs for each file (FIG. 92). These are either the Action IDs from individual files, or the menu shared variable mapped previously.

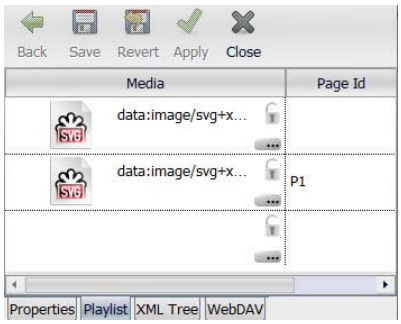


FIG. 92 Adding Page IDs

4. When finished, press the **Save** button at the top of the *Edit Mode* workspace to save your changes.

Testing the Project

Now is time to test the project and make sure that the menu works correctly. To start:

1. In the *Animation Controls* toolbar (FIG. 93), press the **Interactivity Mode** button.

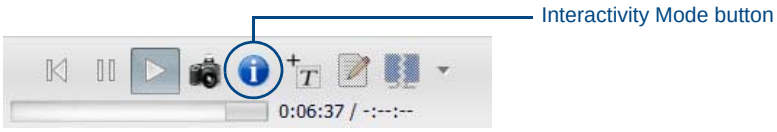


FIG. 93 Animation Controls toolbar

NOTE

When Interactivity Mode is enabled, editing your project is no longer possible. If you need to make corrections to your project, press the Interactivity Mode button again to allow editing.

2. In Interactivity Mode, mouse and keyboard events are now sent to the project. Test your content using the keyboard:
   - Attempt a keystroke from a previously mapped key and observe the result.
   - In Edit Mode, change the Action ID of the keystroke to a variable change, such as NEXT.
   - In Edit Mode, set the selected item menu variable (*menu*) to the configured Page ID.
   - If the event is received, the media file with the corresponding Action ID is enabled and shown.

Suggestions for Further Projects

- Besides the keyboard, XPress projects may be controlled via other inputs. For instance, a project on an IS-SPX Player may be controlled directly with keystrokes through its USB port, or by using the network API (for more information, refer to *Enabling the Network API* on page 88).
- Besides standard button effects, other button effects may be utilized with the help of keystroke mapping. For instance, you could use a keyboard to change a particular displayed text using the news templates, or to play and pause a video file.
- Network configuration options also offer other possibilities, For instance, a project could be constructed with a displayed menu running on one IS-SPX Player and video or other media files running on another. Using the ability to network players (for more information, refer to *Synchronization* on page 52), it is also be possible to control the output of more than one IS-SPX Player via one menu.

Playback Projects

In some circumstances, you may want to use XPress to create media player-like applications. This may include the ability to run introductory videos in an exposition display, or running a video file to augment a particular operation on a keyboard menu. Creating a playback project involves three major steps:

- Controlling the play/pause mode of a media file, using the *playback.svg* file.
- Controlling menu buttons (play, stop, pause, etc.), using the *menu_button.svg* file.
- Showing and controlling the progress of the media file, using the *slider.svg* file.

In this example, the object is to create a basic media player that runs one media file. This file will be controlled with five individual controls:

- **Play** - The media file runs from its beginning to its end.
- **Pause** - The media file stops at the selected point, restarting at that point when play resumes.
- **Stop** - The media file returns to its beginning when restarted.
- **Back** - The media file automatically returns to its beginning.
- **Slider** - The media file may be run forward or in reverse, depending upon where in its scale the slider cursor is located.

Playback Template

The playback.svg file has several functions, including allowing you to play and pause a media file and getting information about the file's running time. To investigate its features:

1. In the *My Projects* tab in the Workspace, right-click anywhere in the Workspace and select *New Project* (FIG. 94).

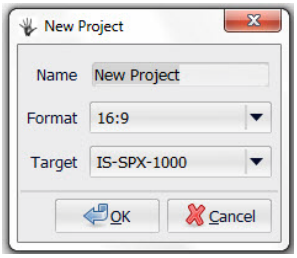


FIG. 94 New Project

2. Enter *Playback* into the *Name* field and click **OK**.
3. Double-click the *index.svg* file to open it, and click the *Layers* tab in the *Edit Mode* Workspace if it is not already open.
4. From the *Interactivity* folder in the *My Templates* tab of the Workspace, drag and drop the *playback.svg* file into the *Layers* tab on the *index.svg* file (FIG. 95).

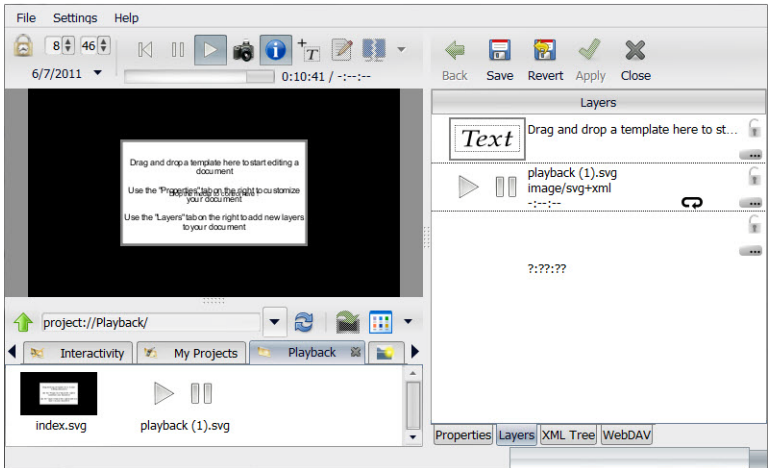


FIG. 95 Playback.svg

5. Double-click on the playback.svg file to open it (FIG. 96).

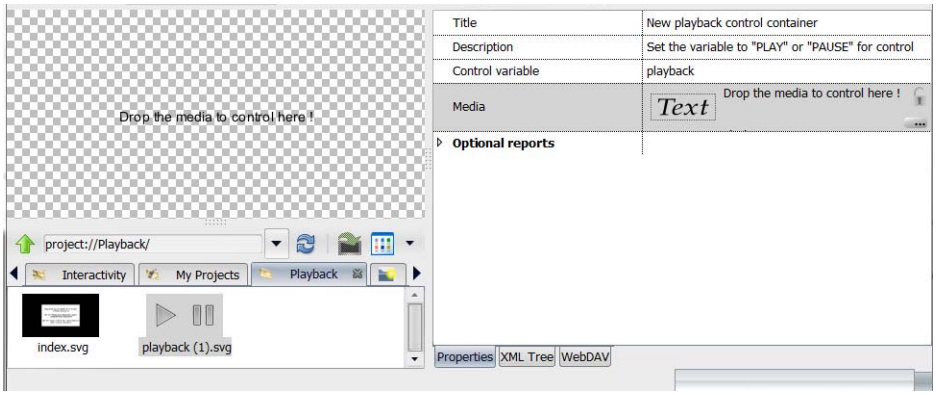


FIG. 96 Playback.svg - Edit Mode Workspace

6. Click on the *Optional Reports* arrow in the *Edit Mode* Workspace to open it (FIG. 97).

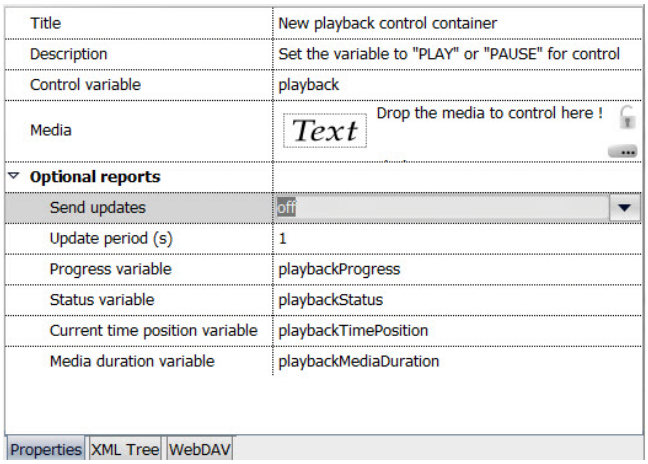


FIG. 97 Optional Reports section

7. The commands supported by the *playback.svg* file are:
   - **PLAY** [time]
   - **PAUSE** [time]
   - **TOGGLE** [time] (this command allows you to switch between PLAY and PAUSE)

     The [time] variable is optional, but allows you to set either a relative time (in percentages) or an absolute (in minutes and seconds) to each command.
8. Click on the *Send Updates* field under *Optional reports* and select *On* from the dropdown menu. The updates must be enabled for proper function of the *playback.svg* file.

Controlling the Media

The PLAY and PAUSE modes can be controlled with several options:

- Using the menu_button.svg file.
- Using keystrokes through the keystroke.svg file.
- Using the network API, via a shared variable modified through the network.

Playback Position

The actual playback position may be controlled using the *slider.svg* file. When added to a project, the slider can show the progress of a media file, and it can also control the position in time in that media file by moving it back and forth. The *slider.svg* file may also be used on text boxes (for particularly long text boxes) and for various custom media.

Example: Media player

For this exercise, you will gather together four different elements:

- Playback control with a media file.
- A menu for the control buttons.
- A slider to show the media file progress.
- Keystroke mapping for keyboard shortcuts.

To begin:

1. In the *Playback* project, double-click on the *playback.svg* file to open it (FIG. 98).

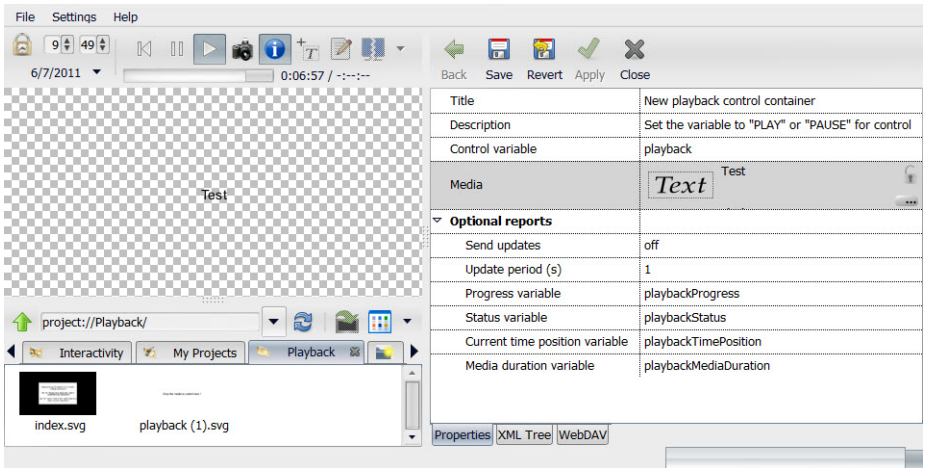


FIG. 98 Playback.svg details

2. Drag and drop a media file into the Workspace. Click the **Save** button at the top of the *Edit Mode* Workspace to save your changes.
3. Double-click the *index.svg* file to open it.
4. Add a background to the project from the selection in the *My Templates* tab and expand it across the Workspace (FIG. 99).

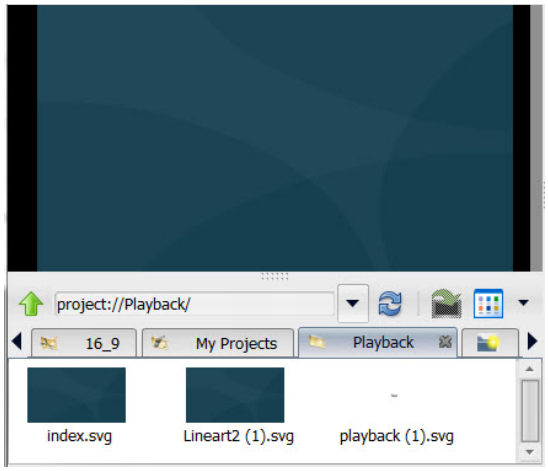


FIG. 99 Added background

5. Add the *slider.svg* file to the project and place it near the bottom of the Workspace (FIG. 100).

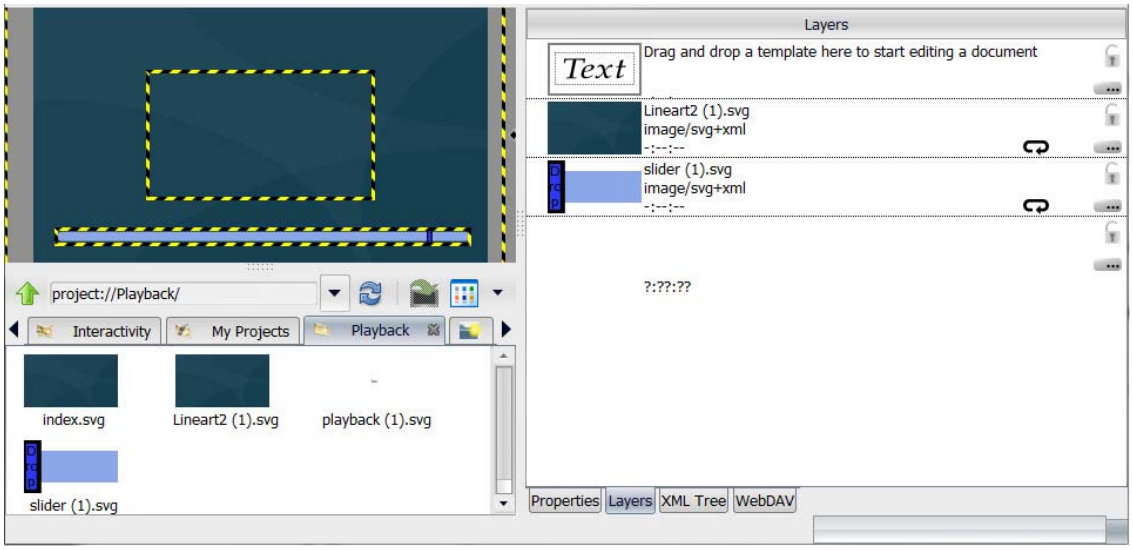


FIG. 100 Slider.svg placement

6. Double-click the *slider.svg* file and view its properties (FIG. 101). Make any changes you wish to the color or the text and click the **Save** button to save your changes.

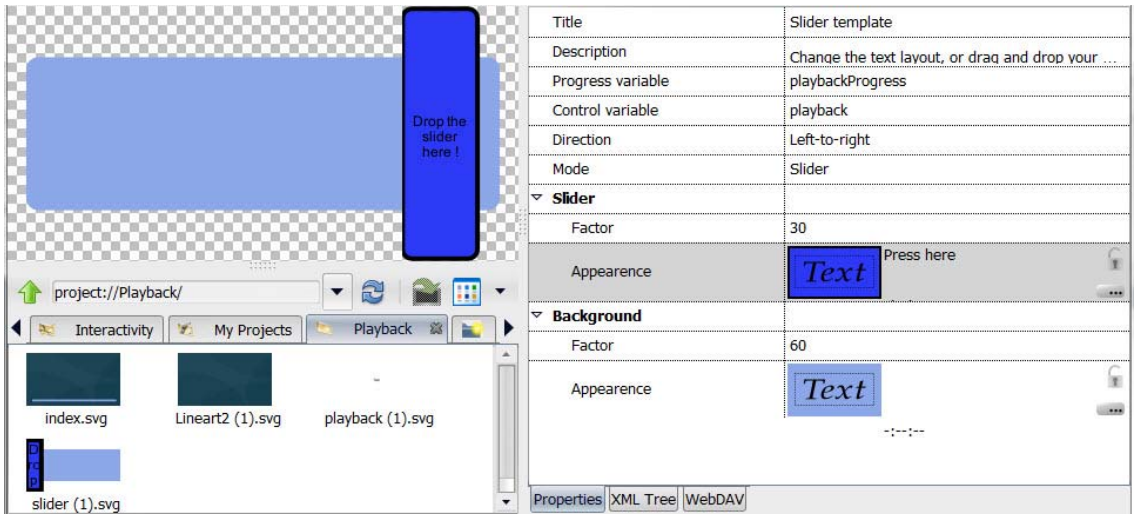


FIG. 101 Slider.svg properties

7. Add menu buttons (as in the *Creating Buttons* on page 64), configure the settings on the menu with the *playback.svg* file (as in the *Controlling Media* on page 72), and set keyboard shortcuts (as in the *Keystrokes* on page 71).

8. When you are finished, test the project by clicking the **Interactivity Mode** button (FIG. 102) and checking the results.

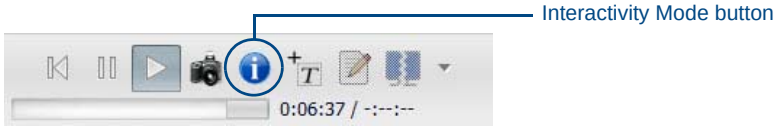


FIG. 102 Interactivity Mode button

IS-SPX Player Configuration

In order to run interactive projects, the IS-SPX Player must be configured to run them. This requires access to the player's Browser-Based Configuration Pages.

NOTE

For more information on the Browser-Based Configuration Pages, please refer to the IS-SPX-1000 & IS-SPX-1300 Operation Reference Guide, available at ***www.amx.com****.*

To start, events must be enabled on the IS-SPX Player. To do this:

1. From the Browser-Based Configuration Pages, select *Display Settings*.
2. In the *Display Settings* page, select the *Interactivity* tab (FIG. 103).

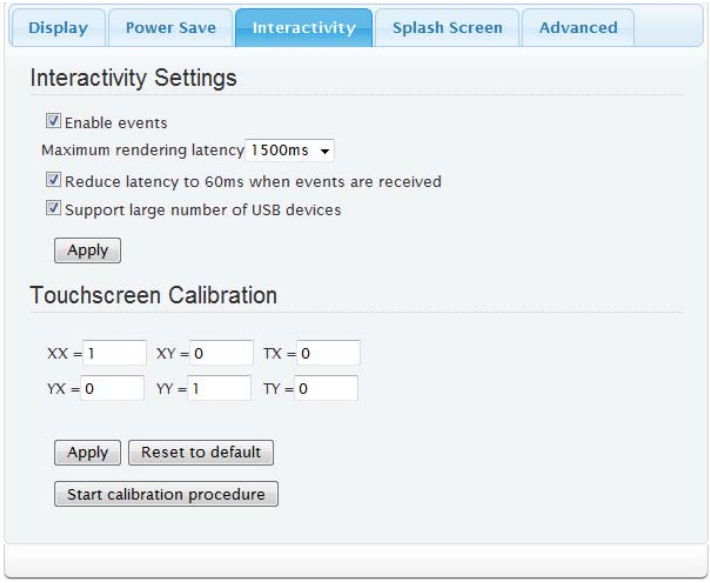


FIG. 103 Interactivity Settings

3. If the *Enable events* checkbox is not selected, select it now.
4. Select your *Maximum rendering latency* setting. This ranges from 250 milliseconds to 1500 milliseconds.
   - When using audio, use at least a latency of 250 milliseconds. 500 milliseconds is recommended.
5. If you wish to lower your latency levels upon reception of interactivity events, click the *Reduce latency to 60ms when events are received* checkbox. When using audio in a project, do NOT click this checkbox.
6. If you wish to calibrate the touchscreen at this time, please refer to the *Calibration For Touch Screens* on page 79.
7. Next, select **Administration > Network Settings** in the Browser-Based Configuration Pages.
8. Click on the *Advanced* tab (FIG. 104).

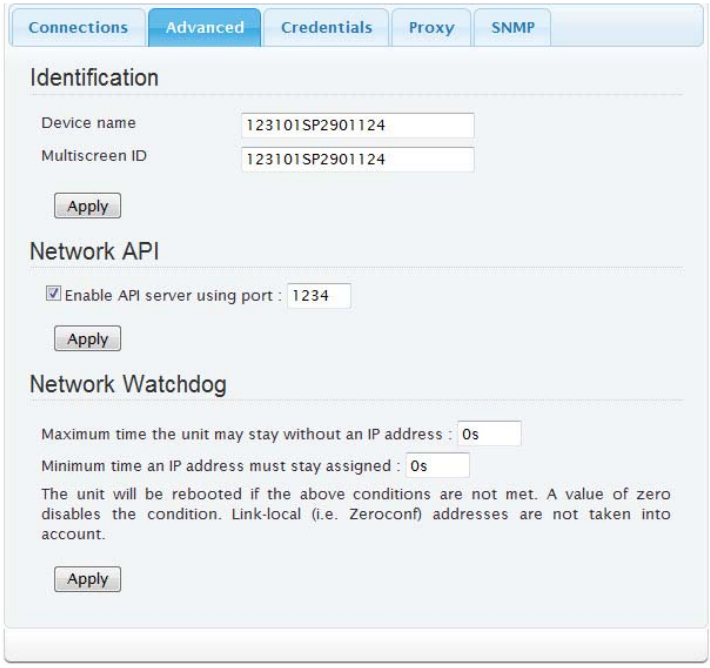

Connections
Advanced
Credentials
Proxy
SNMP
Identification
Device name
123101SP2901124
Multiscreen ID
123101SP2901124
Apply
Network API
Enable API server using port : 1234
Apply
Network Watchdog
Maximum time the unit may stay without an IP address : 0s
Minimum time an IP address must stay assigned : 0s
The unit will be rebooted if the above conditions are not met. A value of zero disables the condition. Link-local (i.e. Zeroconf) addresses are not taken into account.
Apply

FIG. 104 Advanced tab

9. By default, the Network API port is disabled on the IS-SPX Player. Click the *Enable API server* checkbox and then click **Apply**.
10. The IS-SPX Player is now configured to accept and process interactive projects.

Calibration For Touch Screens

From time to time, the touch screen associated with an IS-SPX Player may need calibration to assure proper function. This may be done both through the player, or through the particular touch screen.

Calibration of a Touchscreen Through the IS-SPX Player

Calibration of a connected touchscreen may be done directly from the IS-SPX Player. To start calibration:

1. In the IS-SPX Player's Browser-Based Configuration Pages, select **Administration > Display Settings**.
2. In the *Display Settings* page, select the *Interactivity* tab (FIG. 105).

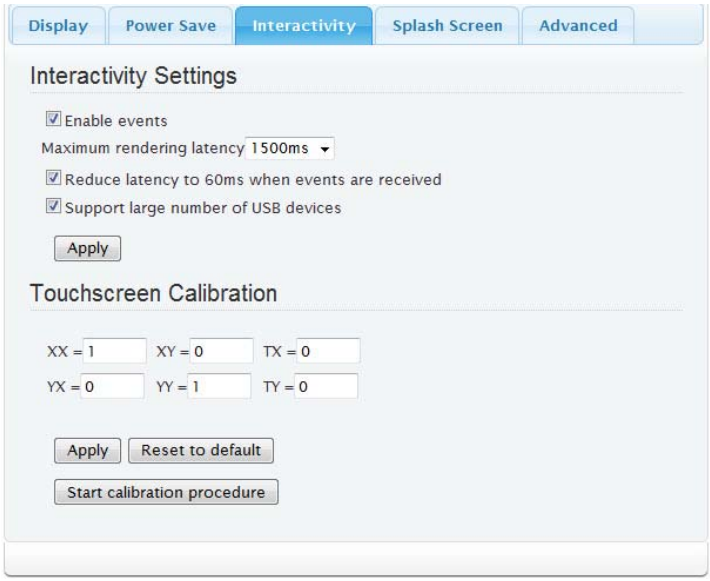


FIG. 105 Interactivity tab

3. Press the **Start calibration procedure** button and follow the calibration procedure on the connected touchscreen. If you need to stop the calibration procedure at any time, press the Stop calibration procedure button (FIG. 106).

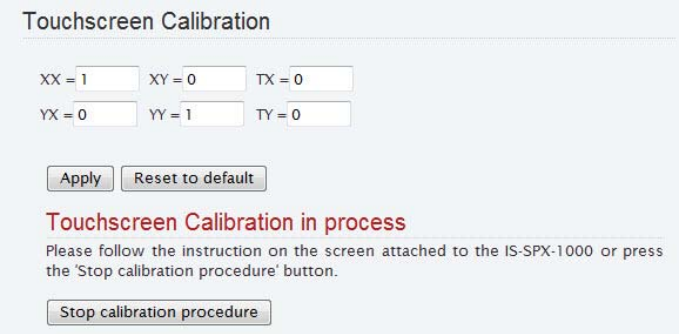

Touchscreen Calibration
XX = 1
XY = 0
TX = 0
YX = 0
YY = 1
TY = 0
Apply
Reset to default
Touchscreen Calibration in process
Please follow the instruction on the screen attached to the IS-SPX-1000 or press the 'Stop calibration procedure' button.
Stop calibration procedure

FIG. 106 Touchscreen Calibration in process

Manual Calibration Procedure

For manual calibration of a touchscreen:

1. Press on the screen to start.
2. Click on the crosshairs as they are presented.
3. Click on the screen to apply the configuration.
4. Once finished with calibration, the normal display will start again.

Troubleshooting

When calibrating a touchscreen through the IS-SPX Player:

- The user interface must NOT be password protected.
- Input events must be enabled.

Serial Protocol File Editor

Overview

The AMX Serial Protocol Editor allows users to create and edit serial protocol files that are compatible with the Inspired Signage XPress range of players. These files facilitate the usage of human-readable commands to be translated into RS232 commands which are then sent to attached devices. The AMX Serial Protocol Editor is available for use at **www.amx.com**.

NOTE

For more information on creating and editing serial protocol files, such as gathering display serial protocol commands and examples of hex string input, refer to Serial Control on page 91.

Before you can start creating the RS232 commands, you must first enter some basic information which will be used later on. These include:

- **Manufacturer** - The manufacturer of the device intended to use the protocol file.
- **Model** - The model of the device.
- **Target** - the type of device (monitor, GPS, NetLinx, etc.).

You cannot proceed without completing these fields (FIG. 107).

FIG. 107 Serial Protocol File Editor

Two types of variables are supported in this release of the Serial Protocol File Editor: *List variables* and *Range variables*. List variables are variables that can have several states, where each state has its own command associated with it. Examples of List variables include *power*, *mute* and *input source*. The *power* and *mute* variables can have a state of either on or off, while *input source* can have states like HGMI, VGA and D-VIDEO. *Volume*, which has a range of 0 to 100, would be a Range variable. Both types of variables are created by clicking the **Add Variable** button.

Adding a List Variable

Clicking **Add Variable** will display the New Variable pane (FIG. 108). Within this pane, enter a Variable Name, then select *List*. To proceed, click **Next**.

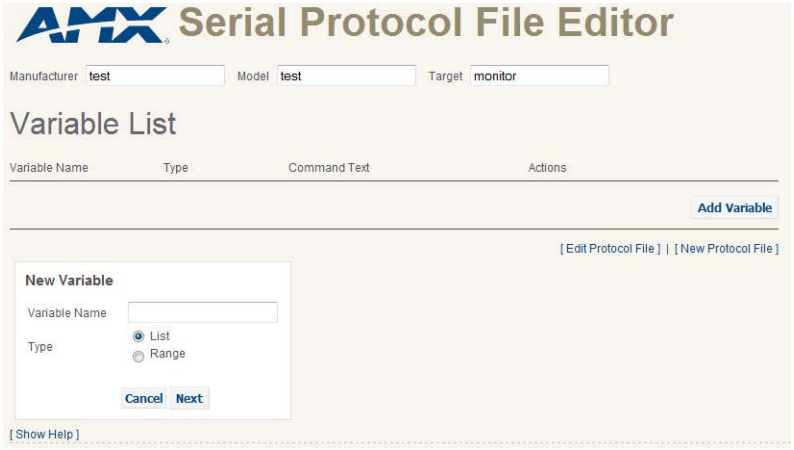


FIG. 108 New variable pane

NOTE

Shared variables take the format amx:control:[TARGET]:[VARIABLE]. The combination of Target and Variable is used when creating the shared variable to create the human readable format of the commands. For example, a Target of "monitor" and a variable named "power" would create a shared variable named "amx:control:monitor:power" that you can then use to control the attached device.

In the *Edit Variable* page (FIG. 109), enter a Command Name, select whether the command is going to be entered as Text or Hex, and enter the appropriate command text.

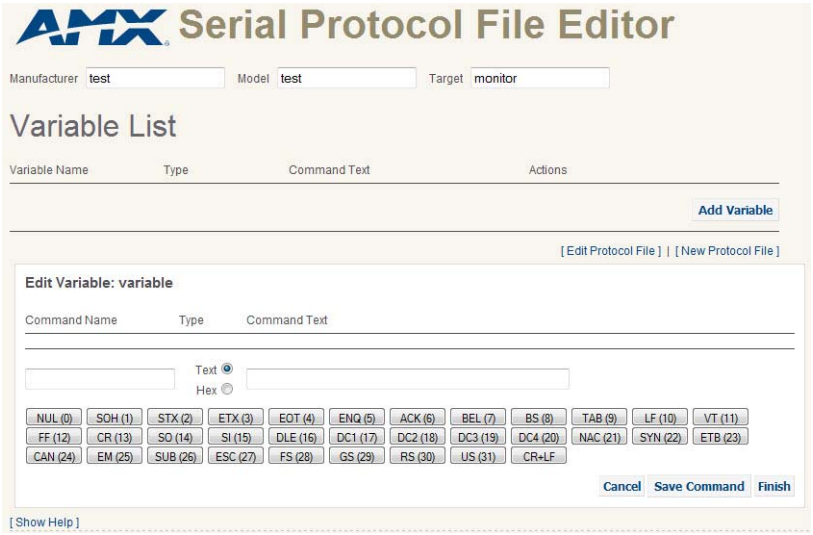

Serial Protocol File Editor
Manufacturer test
Model test
Target monitor
Variable List
Variable Name
Type
Command Text
Actions
Add Variable
[Edit Protocol File] | [New Protocol File]
Edit Variable: variable
Command Name
Type
Command Text
Text
Hex
NUL (0) SOH (1) STX (2) ETX (3) EOT (4) ENQ (5) ACK (6) BEL (7) BS (8) TAB (9) LF (10) VT (11)
FF (12) CR (13) SO (14) SI (15) DLE (16) DC1 (17) DC2 (18) DC3 (19) DC4 (20) NAC (21) SYN (22) ETB (23)
CAN (24) EM (25) SUB (26) ESC (27) FS (28) GS (29) RS (30) US (31) CR+LF
Cancel
Save Command
Finish
[Show Help]

FIG. 109 Edit Variable Page - List variable

You will notice that below the Command fields are 3 rows of buttons. These buttons allow you to enter special characters in both Text and Hex modes. You will notice that depending on the mode, the values associated with each button press is specially encoded. This is to avoid security issues associated with non-standard HTML characters. These specially encoded characters will be decoded correctly when the FSM file is generated.

When finished, click **Save Command** if you would like to add more commands to this list, or **Finish** if you are done with the list and want to move on to the next variable, or you wish to export the completed FSM file.

Adding a Range Variable

As with adding a List variable, clicking **Add Variable** will display the New Variable pane. Within this pane, you must enter a Variable Name, then select *Range*. Selecting Range causes the *From* and *To* fields to appear (FIG. 110), where you can enter the lower and upper limit of this variable. Clicking **Next** will again take you to the *Edit Variable* page.

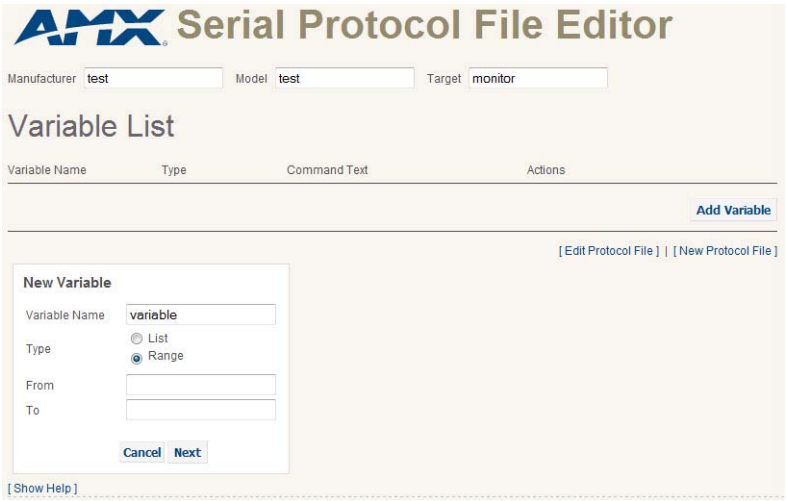


FIG. 110 Range Variable pane

When using the *Edit Variable* page for a Range variable, the *Command Name* field is no longer available. This is because for this type of variable, the emphasis is on the contents in the *Command Text* field. As with a List variable, select to enter the command itself, using either Text or Hex.

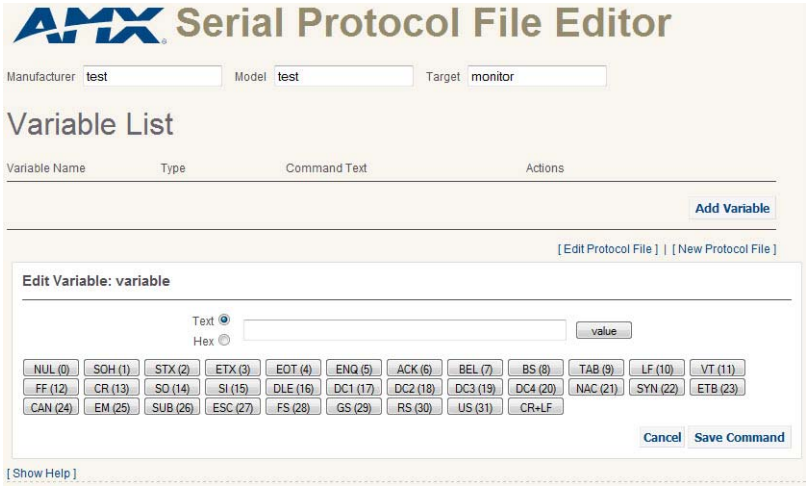

Serial Protocol File Editor
Manufacturer test
Model test
Target monitor
Variable List
Variable Name
Type
Command Text
Actions
Add Variable
[Edit Protocol File] | [New Protocol File]
Edit Variable: variable
Text
Hex
value
NUL (0) SOH (1) STX (2) ETX (3) EOT (4) ENQ (5) ACK (6) BEL (7) BS (8) TAB (9) LF (10) VT (11)
FF (12) CR (13) SO (14) SI (15) DLE (16) DC1 (17) DC2 (18) DC3 (19) DC4 (20) NAC (21) SYN (22) ETB (23)
CAN (24) EM (25) SUB (26) ESC (27) FS (28) GS (29) RS (30) US (31) CR+LF
Cancel
Save Command
[Show Help]

FIG. 111 Edit Variable page - Range variable

Due to the special nature of Range variables, and to avoid creating huge lists of constants, users can have the value being passed into this variable directly inserted into the Command Text at runtime. This is done by specifying the #value# tag within the *Command Text* field. For instance, the LG command *kf 00 #value#$OD;$0A;* shows that the value 0 to 100 would be placed inside this command text when the command is executed. Also, the value button automatically inserts the *#value#* tag at the current cursor location when clicked.

Exporting the Finished File

When you have completed your variable lists, make sure to click **Save Command** with the last variable, and then click **Finish** to save your list. This both shows the variable list and enables the ability to export the list as a file usable by the IS-SPX Player (FIG. 112).

FIG. 112 Variable List with Export link

To save the file as an FSM file, click **Export XPress File** and select **Save File**. From here, the file may be uploaded to an IS-SPX Player or transferred to another tool for further editing.

Shared Variables

Overview

The IS-SPX Player introduces a new mechanism called “shared variables”. This mechanism is based on a client/server model whereby a server maintains a collection of variables. The value of the variables may be updated at any time. When needed, one or more clients connect to the server to get the current value of the variables. The client can further elect to receive real-time notification when the value of those variables is updated.

The client and the server might be located on the same physical host. In this case, the shared variables are used to share information within documents. For instance, this can be used to generate interactive menus. The server might be a remote PC or another IS-SPX Player. In this case the shared variables can be used to trigger many types of events on the IS-SPX Player, such as changing the content displayed by the player, displaying a message on the screen, or the real time update of news feeds.

Concepts Behind Shared Variables

The operation of notifying the server of the list of variables for which notification is desired is called a “subscription”. Clients may subscribe and un-subscribe to variables at any time. When a client first subscribes to a variable and this variable already has a value, it will immediately receive a notification with the current value. Subsequently, the client will receive notifications each time the value has been updated, regardless of whether the value is actually different from what it was before.

The update notification always carries an absolute UTC timestamp that may be used to synchronize the action of multiple clients on a variable update. This publish and subscribe mechanism resembles the “messaging” feature of enterprise application frameworks with the exception that due to the real-time orientation only the most up to date value of a variable is distributed, so some notifications may be dropped if a newer update is available.

It is possible to subscribe to input events (mouse clicks or keyboard inputs) from an IS-SPX Player. In this case, all events received by the server will be passed to the client. For instance, it is possible using a keyboard plugged into an IS-SPX Player to control not only the attached player but also other players connected through the network. The special subscription variable “%EVENTS%” will cause event notifications to be sent instead of updates.

Javascript API

Shared variables are meant to be used with the JavaScript and NetLinx programming languages.

Interface

```
interface SharedVariable {
readonly attribute DOMString name;
readonly attribute DOMString value;
readonly attribute float lastUpdateTime; // milliseconds since Jan
1st 1970 UTC
void set( DOMString newValue );
void addUpdateListener( UpdateListener listener );
void removeUpdateListener( UpdateListener listener );
};
void UpdateListener( SharedVariable var );
interface SVGGLobal {
SharedVariable createSharedVariable( DOMString name );
};
```

The main API is the global createSharedVariable function, which is used both to create a local variable and to connect to a remote one.

A local variable will be stored in a non-persistent manner on the IS-SPX Player. This means that a reboot of the IS-SPX Player will reset all shared variables to empty values. Local variables can be accessed from remote clients only when the network API of the player is enabled.

To connect to a remote variable, the name parameter must contain the variable name followed by '@' and the hostname of the variable server. Optionally, a TCP port should be added if the port number on the server is different from the port number on the local host, e.g: *text@remotehost:4567*.

The function *createSharedVariable* returns a SharedVariable object that should be used to install “listeners”. Those are callbacks that will be called whenever the variable value changes.

The callback is passed a reference to the variable object and can query the value property. It is also possible to poll on the *lastUpdateTime* property (initially *0*) to detect when the value has been updated. The set method is used to update the value of a variable. It will trigger any callback installed on the same variable object or on other variable objects created with the same name, including remote objects if networking is enabled.

Sample Code

Setting the Variable

On the player that controls the variable

```
var v=createSharedVariable( 'XY' );
v.set( 'new value' );
```

Reading the Variable

When the same player sets and read the variable:

```
function onUpdate( x ) {
// Do something based on x.value
}
var v=createSharedVariable( 'XY' );
v.addUpdateListener( onUpdate );
```

When a player sets the variable and another player reads its content:

```
function onUpdate( x ) {
// Do something based on x.value
}
var v=createSharedVariable( 'XY@controller' );
v.addUpdateListener( onUpdate );
```

Network API

Framing

The network protocol for the distribution of notifications is based on a stream of variable length commands sent over a TCP socket. Individual commands are separated by the \r\n *(0x0D 0x0A)* two-byte sequence. This sequence should not happen inside the command. When string parameters are used, they are put between double quotes. Special characters should be escaped, in a manner similar to C-string constants, with a backslash sequence. The following characters must be escaped:

- newline (*0x0A*) with \n
- carriage return (*0x0D*) with \r
- double quote (*0x22*) with \"
- null byte (*0x00*) with \0
- backslash itself with \\

Alternatively, the network API also supports setting shared variables via an HTTP GET request. This method does not allow persistent connections, but it is easier to access from a client scripting language supporting the XMLHttpRequest API, like JavaScript in web browsers.

Whenever a GET request is issued, the connection by the server is closed immediately after the response has been sent.

The HTTP GET interface supports only the 'UPDATE' and 'EVENT' commands.

List of Commands

Network API Commands	

UPDATE	Syntax:
	`UPDATE [timestamp] “name” “value”\r\n`
	Used to send or receive updates. The timestamp is the number of microseconds since Jan 1st 1970 UTC. The timestamp may be omitted, in which case the server will use the time the command is received as the timestamp. It is only possible to send an 'UPDATE' command to an IS-SPX Player if the network API is enabled. Note that no response is sent back by the server. Please note that the protocol is case sensitive, so the word UPDATE must be in capital letters.
	Example:
	`GET / UPDATE?name1=value1&name2=value2... HTTP/1.0\r\n`
	`\r\n`
	Used to send an update for one or more shared variables. The URL is also case sensitive, but the update must be lower case in this instance. It is not possible to specify a timestamp with this interface. Instead of opening a TCP socket and managing the low level protocol, you can get a web browser to send the request by entering the address below:
	`http://[server]:[port]/update?name1=value1&name2=value2...`

EVENT	Syntax:
	`EVENT [timestamp] “name” “value”\r\n`
	Used to send or receive UI events. The timestamp is optional. The name must be one of the UI trigger names:
	• “**keydown**”, “**keyup**”: the value is an SVG key identifier, e.g. “Enter”, a character, e.g. “A” or an hexadecimal unicode character, e.g. “U+0041”; it may be prefixed by one or more of the following modifiers: “Ctrl+”, “Shift+”, “Alt+”, “Meta+”, “AltGr+”, e.g. “Shift+A”.
	• “**textInput**”: the value is the text string input by the user, utf-8 encoded.
	• “**mousedown**”, “**mouseup**”, “**mousemove**” or “**mousewheel**”: the value is a comma-separated list of the following argument: “x”, “y”, “button” and “delta” as applicable, e.g. for a mousedown on the left button at coordinates 45,20, the value string is “x=45,y=20,button=0”
	• “**focusin**” and “**focusout**”: the value must be empty.
	It is only possible to send an 'EVENT' command to an IS-SPX Player if the network API is enabled.
	Example:
	`GET / event?name1=value1&name2=value2... HTTP/1.0\r\n`
	`\r\n`
	Used to send one or more events.
	Instead of opening a TCP socket and managing the low level protocol, you can get a web browser to send the request by entering the address below:
	`http://[server]:[port]/event?event_name1=value1&event_name2=value2...`

SUBSCRIBE	Syntax: `SUBSCRIBE “name”\r\n` Subscribe to a variable on this server (the @ sign and hostname must be removed). The subscriber will be notified each time the variable is modified. If the client subscribes to a variable, it will received an 'UPDATE' command each time the variable is modified. If the client subscribe to the special variable “%EVENTS%”, it will receive an 'EVENT' command each time an event is generated on the server, i.e. each time a keyboard or a mouse input is received on the server. It is only possible to send an 'SUBSCRIBE' command to an IS-SPX Player if the network API is enabled.
UNSUBSCRIBE	Syntax: `UNSUBSCRIBE “name”\r\n` Unsubscribe to a variable.

Sample Code	

Updating a variable using PHP	`function send_update( $host, $port, $name, $value ) {`
	`$s=fsockopen( $host, $port );`
	`fwrite( $s, “UPDATE \”$name\” \”$value\”\r\n” );`
	`fclose( $s );`
	`}`
	Note: this simplified example does not handle the escaping of control characters.

Sending an event using PHP	Sending a keydown event for the key 'V' with control modifier pressed.
	`function send_copy_key( $host, $port ) {`
	`$s=fsockopen( $host, $port );`
	`fwrite( $s, “EVENT \”keydown\” \”Ctrl+V\”\r\n” );`
	`fclose( $s );`
	`}`

Updating a variable using telnet	`telnet 192.168.1.10 1234`
	`Trying 192.168.1.10...`
	`Connected to 192.168.1.10`
	`Escape character is '^]'.`
	`UPDATE “menu” “P1”`
	`^]`
	`telnet> quit`
	`Connection closed.`

Updating a variable using HTTP	`http://192.168.1.10:1234/update?menu=P1`

Enabling the Network API

The network API must be enabled on the IS-SPX Player before it can be utilized. This setting is controlled from the IS-SPX Player Browser-Based Configuration Pages (FIG. 113):

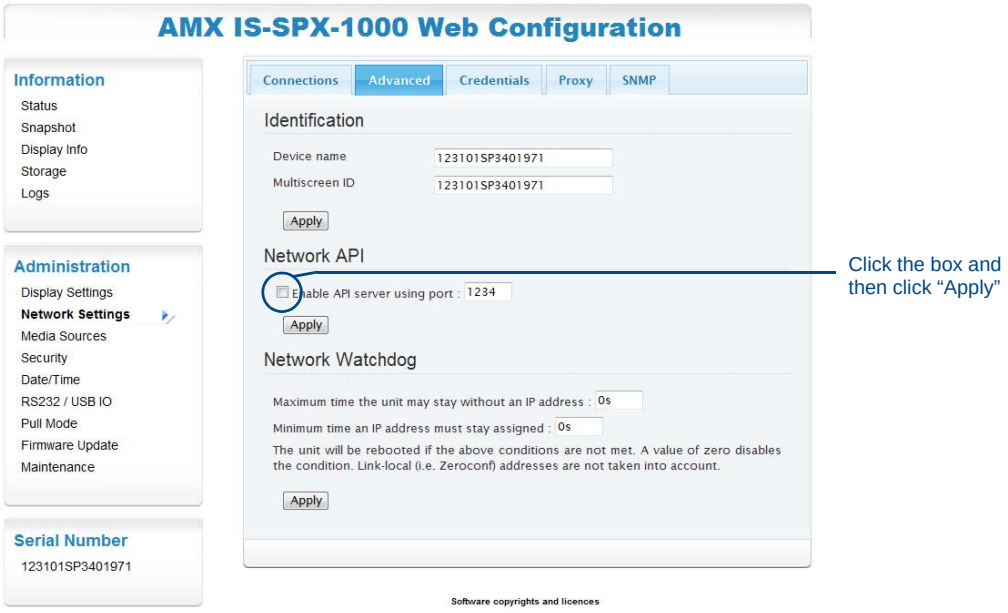


FIG. 113 Enabling the Network API

NOTE

For more information on the Browser-Based Configuration Pages, please refer to the IS-SPX-1000 & IS-SPX-1300 Operation Reference Guide*, available at* ***www.amx.com****.*

API Emulation

XPress may emulate the IS-SPX Player network API on the API port 1234. By default, the API is turned off. The network API is enabled via *Settings / Enable network API* (FIG. 114).

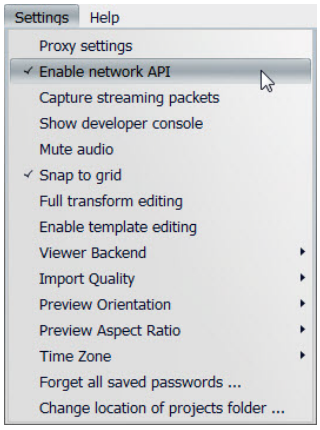


FIG. 114 Enabling the Network API on the Settings menu

Once the network API has been enabled, it is possible for remote IS-SPX Players to listen to shared variables and mouse/keyboard events generated by XPress.

NOTE

Restarting XPress will reset all the shared variables.

Appendix

Serial Control

This section will assist with showing you how to update both the IS-SPX Player and an existing Inspired Signage XPress project to include serial control through the XPress project.

Serial Control From Within an Existing Inspired Signage Project

For this project, you will create an example of serial control through an existing XPress project. In this case, control is through the XPress project and not via NetLinx through the IS-SPX Player. This includes creation of the protocol file and using an LG TV to implement. The intent is to control:

- Power off at a given time (#1)
- Power on at a given time (#2)

Gathering Display Serial Protocol Commands

First, find the appropriate monitor's or display's serial protocol commands. These will be stored within an IS-SPX Player protocol file using the *Serial Protocol File Editor* tool (FIG. 115). This tool is available at **www.amx.com**

FIG. 115 Serial Protocol File Editor

For instance, the Serial Protocol commands below are for an LG TV, model 32LE5300.

Serial Protocol Commands: Text	
Power On:	ka 00 01[cr]
Power Off:	ka 00 00[cr]
Volume (0-100):	kf 00 #value#[cr]
Screen Mute On:	kd 00 01[cr]
Screen Mute Off:	d 00 00[cr]
Volume Mute On:	ke 00 00[cr]
Volume Mute Off:	ke 00 01[cr]
Input Select AV1:	xb 00 20[cr]
Input Select Component:	xb 00 40[cr]
Input Select RCB-PC:	xb 00 60[cr]
Input Select HMDI1:	xb 00 90[cr]
Input Select HMDI2:	xb 00 91[cr]
Input Select HMDI3:	xb 00 92[cr]
Input Select HMDI4:	xb 00 93[cr]

NOTE

*While creating a variable list through the Serial Protocol File Editor, do **NOT** leave the page for any reason until you save your information. If you do so, all previously entered information will be lost.*

Serial Protocol Commands: Hex	
Power On:	6b61203030203031[cr]
Power Off:	6b61203030203030[cr]
Volume (0-100):	6b6620303020#value#[cr]
Screen Mute On:	6b64203030203031[cr]
Screen Mute Off:	6b64203030203030[cr]
Volume Mute On:	6b65203030203030[cr]
Volume Mute Off:	6b65203030203031[cr]
Input Select AV1:	7862203030203230[cr]
Input Select Component:	7862203030203430[cr]
Input Select RCB-PC:	7862203030203630[cr]
Input Select HMDI1:	7862203030203930[cr]
Input Select HMDI2:	7862203030203931[cr]
Input Select HMDI3:	7862203030203932[cr]
Input Select HMDI4:	7862203030203933[cr]

Creating a Serial Protocol File

Next, create a IS-SPX Player protocol file using the AMX Serial Protocol File Editor Web-based tool. Prior to creating the serial commands, three pieces of data are required:

- The monitor or screen's Manufacturer
- The monitor or screen's Model
- The Target for the IS-SPX Player's output

The *Manufacturer* and *Model* information is used in creating the individual file name. *Target* becomes a parameter used when creating serial command Scalable Vector Graphics (SVGs). In this case:

- Set *Manufacturer* to **LG**.
- Set *Model* to **32LE5300**.
- Set *Target* to **monitor**.

Below are the command descriptions followed by an ASCII representation and a hex representation. Either one will work when creating the protocol file. Setting the commands that will be used by the IS-SPX Player is all that is required.

Creating the Power variable:

1. After entering the *Manufacturer*, **Model**, and *Target* information, select *Add Variable* to open the *New Variable* panel (FIG. 116).

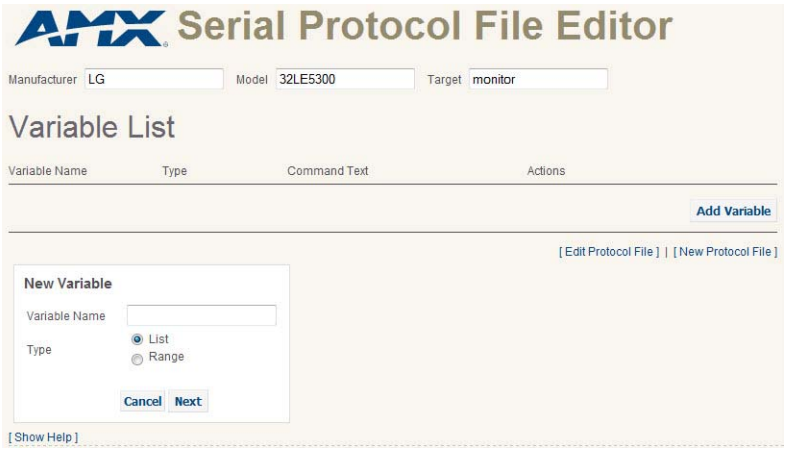


FIG. 116 New Variable panel

2. In the *Variable Name* field, enter the variable's name. In this case, enter "power".
3. From the *Type* radio buttons:, select *List* and then press the **Next** button.

NOTE

If you press Next *without entering a variable name, the page will display "Invalid Command Name".*

4. Enter the *Command Name* in the appropriate field (FIG. 117) and select the *Type* of data to be accepted (*Text* or *Hex*).

Serial Protocol File Editor
Manufacturer LG
Model 32LE5300
Target monitor
Variable List
Variable Name
Type
Command Text
Actions
Add Variable
[Edit Protocol File] | [New Protocol File]
Edit Variable: power
Command Name
Type
Command Text
Text
Hex
NUL (0) SOH (1) STX (2) ETX (3) EOT (4) ENQ (5) ACK (6) BEL (7) BS (8) TAB (9) LF (10) VT (11)
FF (12) CR (13) SO (14) SI (15) DLE (16) DC1 (17) DC2 (18) DC3 (19) DC4 (20) NAC (21) SYN (22) ETB (23)
CAN (24) EM (25) SUB (26) ESC (27) FS (28) GS (29) RS (30) US (31) CR+LF
Cancel
Save Command
Finish
[Show Help]

FIG. 117 Edit Variable panel

5. Below the *Command Name* and *Command Text* fields are 33 buttons, signifying common commands. Depending upon whether you selected *Text* or *Hex*, clicking any of these buttons will enter the appropriate command text into the *Command Text* field. Since the On and Off commands are not within these buttons, they must be entered manually:

Manual Command Text Entries:		
Command Name	**Command Text (Text)**	**Command Text (Hex)**
On	ka 00 01$0D;	6b612030302030310D
Off	ka 00 00$0D;	6b612030302030300D

6. When finished entering the Command text, click **Save Command**. The new command appears in the *Edit Variable* pane (FIG. 118);

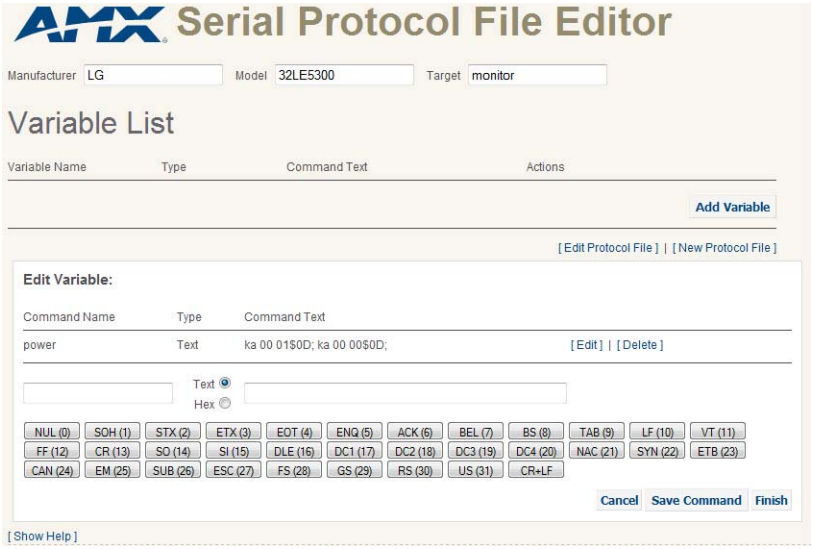


FIG. 118 Variable List

7. Next, click *Add Variable* to enter the *Volume Control* variable (FIG. 119). This time, select *Range* instead of *List*.

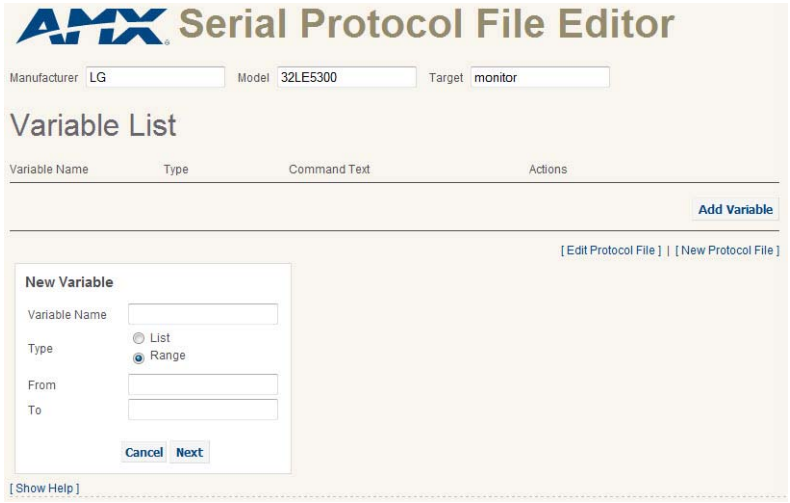


FIG. 119 New Variable - Range

8. Enter the information for the *Volume Control* variable, clicking **Next** to open the next set of entry fields. When finished, click **Save Command**.

Volume Control Variable	
Enter Variable Name:	volume
Select Type:	Range
Enter From:	0
Enter To:	100
Variable as Text:	kf 00 #value#$0D;
Variable as Hex:	6b6620303020#value#0D

9. Next, add the next nine variables, in the same manner:

Screen Mute (Picture On)	
Enter Variable Name:	screenmute
Select Type:	List
Command Name:	On
Variable as Text:	kd 00 01$0D
Variable as Hex:	6b642030302030310D

Screen Mute (Picture Off)	
Enter Variable Name:	screenmute
Select Type:	List
Command Name:	Off
Variable as Text:	kd 00 00$0D;
Variable as Hex:	6b642030302030300D

Mute On (Sound)	
Enter Variable Name:	volumemute
Select Type:	List
Command Name:	On
Variable as Text:	ke 00 00$0D;
Variable as Hex:	6b652030302030300D

Mute Off (Sound)	
Enter Variable Name:	volumemute
Select Type:	List
Command Name:	Off
Variable as Text:	ke 00 01$0D;
Variable as Hex:	6b652030302030310D

Select Input Source for Display	
Enter Variable Name:	inputselect
Select Type:	List: • AV1 • Component • RGB-PC • HDMI1 • HDMI2 • HDMI3 • HDMI4
Command Name:	AV1
Variable as Text:	xb 00 20$0D;
Variable as Hex:	78622030302032300D
Command Name:	Component
Variable as Text:	xb 00 40$0D;
Variable as Hex:	78622030302034300D
Command Name:	RGB-PC
Variable as Text:	xb 00 60$0D;
Variable as Hex:	78622030302036300D

Select Input Source for Display (Cont.)	
Command Name:	HDMI1
Variable as Text:	xb 00 90$0D;
Variable as Hex:	78622030302039300D
Command Name:	HDMI2
Variable as Text:	xb 00 91$0D;
Variable as Hex:	78622030302039310D
Command Name:	HDMI3
Variable as Text:	xb 00 92$0D;
Variable as Hex:	78622030302039320D
Command Name:	HDMI4
Variable as Text:	xb 00 93$0D;
Variable as Hex:	78622030302039330D

10. When you are finished entering the variables, click **Finish** to save your changes. This opens the *Variable List* page, displaying all of your saved variables (FIG. 120).

FIG. 120 Variable List page

11. The next step is to export your serial protocol file by clicking **Export XPress File**. When prompted, click **Save** to save the file to your computer.

Uploading the Serial Protocol File

At this point, you need to upload the file to the IS-SPX Player that will be controlling the monitor in question. This will require access to the IS-SPX Player's Browser-Based Configuration Pages. To upload the file:

1. Under the *Administration* section, select *RS232 / USB IO*.
2. Under *Upload Protocol File*, enter the path for the desired serial protocol file. If necessary, click **Browse** to find the file.
3. After selecting the file, click **Upload** to upload it to the IS-SPX Player. The IS-SPX Player will automatically reboot.

Creating New Templates

The ultimate step will be to create new PowerOn and PowerOff templates in XPress. To start:

1. In XPress, open the *My Templates* folder in the *Workspace* tab (FIG. 121).

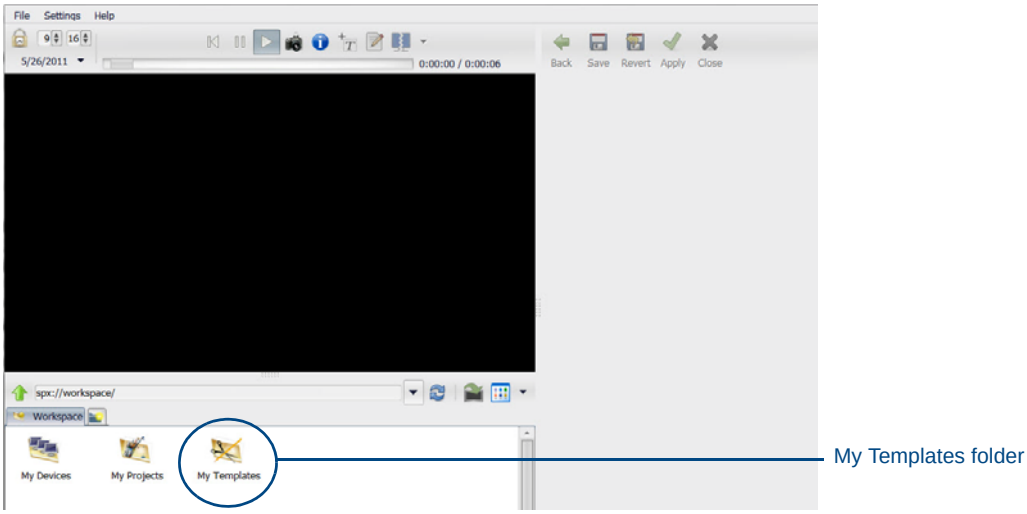


FIG. 121 XPress - My Templates Folder

2. Open the *Serial Port* template folder. In the *Serial Port* folder, click on the *power.svg* file to open it (FIG. 122).

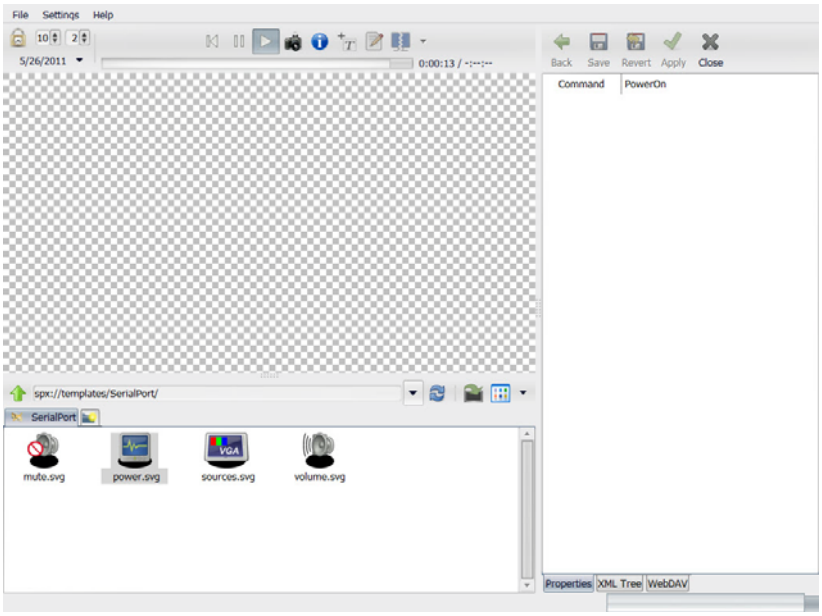


FIG. 122 Power.svg file

3. Right-click the *power.svg* file and copy it. Open your project folder in XPress and paste it in the project window.
4. Right-click the new file and rename it *PowerOn.svg*.
5. Repeat steps 2 through 4 to create *PowerOff.svg* in the same project.
6. Click the *XML Tree* tab (FIG. 122) in *PowerOn.svg* to open the XML command tree.
7. Click the arrow next to the <*auxCMD*> section to open the section and access its contents.
8. Right-click the <*auxCMD*> header and select *Add attribute*. Name the attribute “param1” and press **Enter** to save it (FIG. 123).

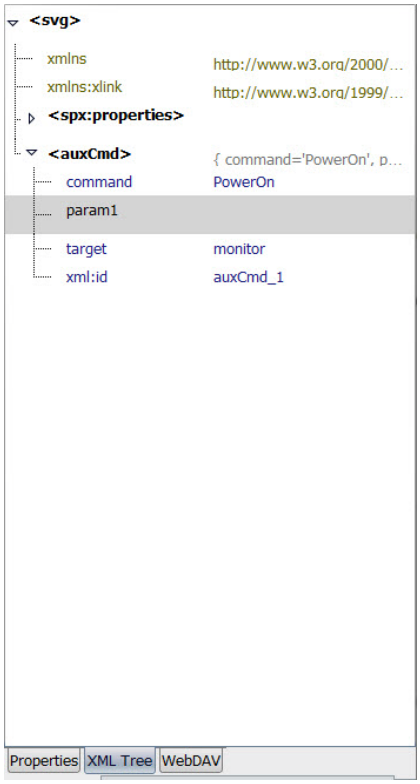


FIG. 123 XML Tree with new attribute param1

9. For *PowerOn.svg*, click on *command* and then select the attribute to the right (currently "PowerOn"). Change the command to *power* and press **Enter** to save it.
10. Click on *target* and change it to *monitor*.
11. Click on *param1* and enter *On*.
12. Repeat steps 6 through 11 for *PowerOff.svg*. In this case, though, set *param1* to "Off".

Creating and updating a new calendar file

The final step involves setting the two new svg files so they can be accessed by the project. To do this:

1. If you have not already done this, copy the PowerOn.svg and PowerOff.svg files into your project.
2. Right-click the workspace area of your project and select *New Schedule*.
3. Open the new calendar file named *new.ics*.
4. Click on the PowerOn.svg file and drag it to the date you want the power to turn on (FIG. 124).

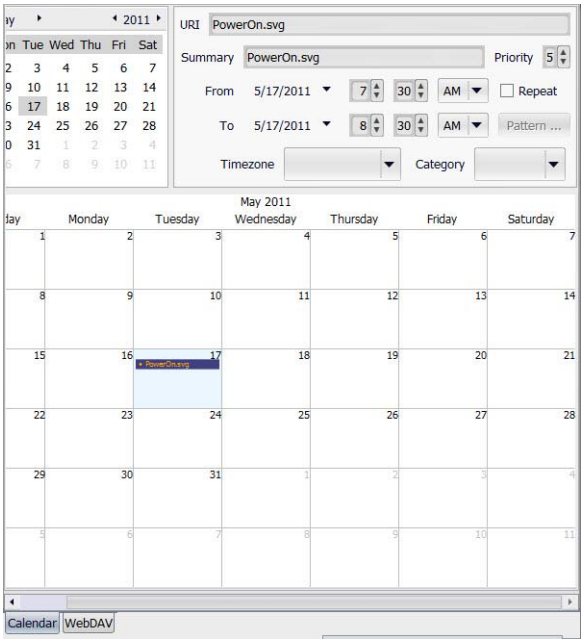


FIG. 124 new.ics file

5. Set the *From* time to the desired time you want the monitor power to turn on.
6. Set the *To* time to one minute after the *From* time.
7. Click the *Repeat* checkbox so this event will happen every day.
8. Click and drag *PowerOff.svg* to the current day in the calendar.
9. Set the *From* and *To* times as in steps 5 through 7, including clicking the *Repeat* checkbox.
10. In the Workspace area, open the *index.svg* file. Click the *Layers* tab if it has not already been selected and drag the *new.ics* file into the *Layers* pane (FIG. 125).

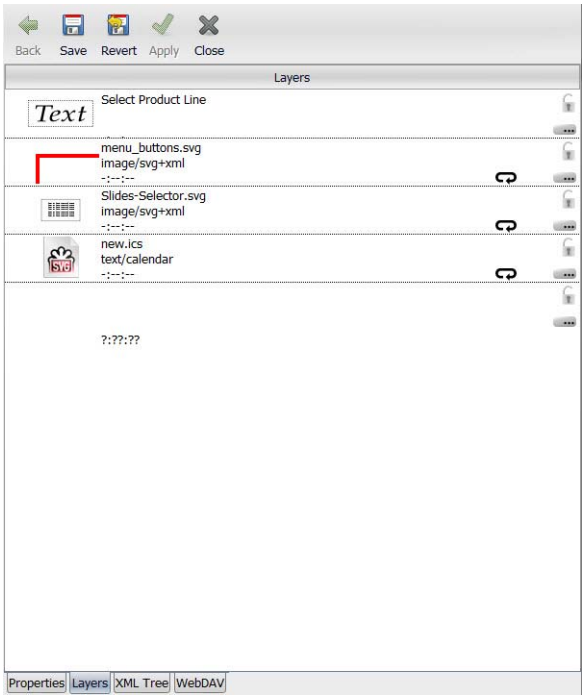


FIG. 125 Layers pane

11. Save the project. The monitor will now power on and off at the time set in the new.ics file.

Serial Control From NetLinx

Serial control in NetLinx is roughly similar to serial control within an XPress project, in that the target parameter used when defining the serial protocol file is also used in the *Send_String* command. It is the value that follows “amx:control:” in the command line. The example below uses a target of "monitor". The variables used in the Send_String command are:

- target
- variable name of command type
- command

The generic NetLinx code would be:

```
SEND_STRING dvIPServer,”'UPDATE “amx:control:{target}:{variable name}”
“{command}”',13,10,13,10”
```

Below is the power command as defined for this monitor.

Power command			
Variable Name	**Command**	**Hex or Text**	**Serial Command**
Power	On	Text	ka 00 01$0D;
Power	Off	Text	ka 00 00$0D;

With these variables, the resultant NetLinx code would read:

```
SEND_STRING dvIPServer,”'UPDATE “amx:control:monitor:power” “Off”',13,10,13,10"
SEND_STRING dvIPServer,”'UPDATE “amx:control:monitor:power” “On”',13,10,13,10”
```

Adding Additional Templates to Inspired Signage XPress

While Inspired Signage XPress contains many pre-loaded templates for building XPress projects, many others are available for download at **www.amx.com**. These include templates for multiscreen content (reference *Synchronized Content* on page 45) and interactivity (reference *Interactivity* on page 63).

After downloading the templates from the AMX web site, you will need to install them in the *My Templates* folder in the XPress directory. For instance, if you made a standard installation of XPress to your computer in Windows 7, the template files will be located at *C:\Program Files\AMX Control Disc\Inspired Signage\XPress\share\templates* (FIG. 126).

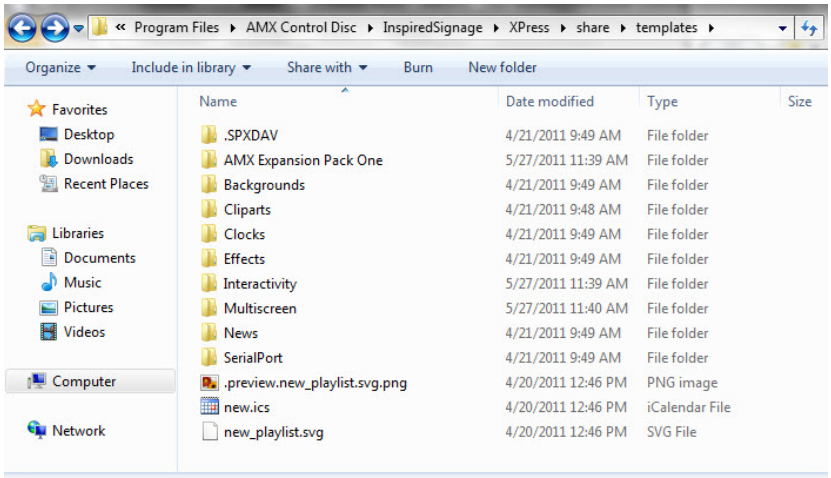


FIG. 126 Directory for XPress templates

After adding the template files to the directory, open XPress. The files should appear in the *My Templates* folder in the Workspace (FIG. 127). If they do not, select *File* > *Refresh* from the menu to refresh the Workspace

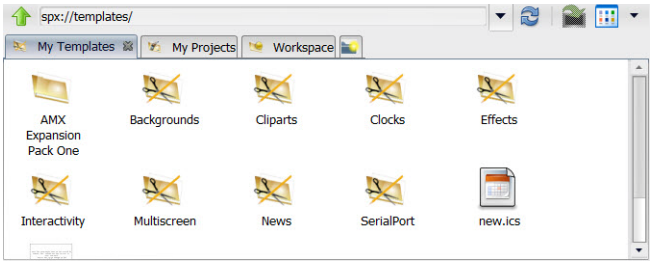


FIG. 127 New templates in the My Templates workspace

NOTE

Individual files and folders may be dragged and dropped from your computer's desktop or Windows Explorer windows into the My Projects tab, but this cannot be done with the My Templates tab. All new templates must be added to the templates *folder before they can appear in the My Templates tab.*

Commonly Used Terms

FSM

The *.fsm* extension on files generated by the Serial Protocol File Editor (for more information, refer to *Serial Protocol File Editor* on page 81) refers to Finite State Machine. A Finite State Machine is a behavior model composed of a finite number of states, as well as transitions between those states and actions, similar to a flow graph in which one can inspect the way logic runs when certain conditions are met.

SVG

Scalable Vector Graphics (SVG) is a family of specifications of an XML-based file format for describing two-dimensional vector graphics, both static and dynamic (i.e. interactive or animated). The SVG specification is an open standard that has been under development by the World Wide Web Consortium (W3C) since 1999. SVG images and their behaviors are defined in XML text files. This means that they can be searched, indexed, scripted and, if required, compressed. Since they are XML files, SVG images can be created and edited with any text editor, but drawing programs are also available that support SVG file formats.

URI

In this document, URI stands for *Uniform Resource Identifier*, This is more specific than the more standard URL (*Uniform Resource Locator*). In some operations, the IS-SPX Player Browser-Based Configuration Pages and the interface of XPress request a URI for network connections. A standard URL may be used in these cases. Check with your network administrator if a specific URI needs to be used instead.

Increase Your Revenue through education + knowledge

In the ever-changing AV industry, continual education is key to success. AMX University is dedicated to ensuring that you have the opportunity to gather the information and experience you need to deliver strong AMX solutions. Plus, AMX courses also help you earn CEDIA, NSCA, InfoComm, and AMX continuing education units (CEUs).

Visit AMX University online for 24/7/365 access to:

- *Schedules and registration for any AMX University course*
- *Travel and hotel information*
- *Your individual certification requirements and progress*

11/13 ©2013 AMX. All rights reserved. AMX and the AMX logo are registered trademarks of AMX. AMX reserves the right to alter specifications without notice at any time.

3000 RESEARCH DRIVE, RICHARDSON, TX 75082 USA • 800.222.0193 • 469.624.8000 • 469-624-7153 fax • 800.932.6993 technical support • www.amx.com